



Miguel dos
Santos Malheiro

Guia turístico pessoal baseado em
contexto através de PDA (servidor)



**Miguel dos
Santos Malheiro**

Guia turístico pessoal baseado em contexto através de PDA (servidor)

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Dr. José Manuel Matos Moreira, Professor Auxiliar Convidado, e do Dr. Joaquim João Estrela Ribeiro Silvestre Madeira, Professor Auxiliar, ambos do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

o júri / the jury

presidente / president

Doutor António Manuel de Brito Ferrari Almeida

Professor Catedrático da Universidade de Aveiro

vogais / examiners committee

Doutor António Fernando Vasconcelos Cunha Castro Coelho

Professor Auxiliar do Departamento de Engenharia Informática da Faculdade de Engenharia da Universidade do Porto

Doutor Joaquim João Estrela Ribeiro Silvestre Madeira

Professor Auxiliar da Universidade de Aveiro (co-orientador)

Doutor José Manuel Matos Moreira

Professor Auxiliar Convidado da Universidade de Aveiro (orientador)

**agradecimentos /
acknowledgements**

Em primeiro lugar quero agradecer ao orientador e co-orientador, os Professores José Moreira e Joaquim Madeira, pelo apoio e incentivo dado ao longo desta dissertação.

A minha família, em especial à minha avó, aos meus pais, irmãos, tia Céu e tio Quintino pelo apoio e carinho que sempre tiveram.

Aos amigos, pelos incentivos, apoio e amizade demonstrados ao longo do curso.

Aos demais professores da Universidade de Aveiro que contribuíram para a minha formação académica.

Muito obrigado!

Palavras-Chave

Serviços baseados na localização, dispositivos móveis

Resumo

O sector do Turismo desempenha um papel importante na economia portuguesa e tem apresentado um crescimento significativo ao longo dos anos. É, assim, um sector muito competitivo e bastante significativo em termos de investimento.

A aplicação das novas tecnologias da informação e comunicação, no sector do Turismo, permite a oferta de serviços adicionais por parte dos operadores turísticos, como resposta às exigências crescentes dos turistas. A crescente tendência de utilização de sistemas móveis de informação turística, combinada com a possibilidade de localização, suscita o desenvolvimento de aplicações que forneçam aos turistas informação e funcionalidades dependentes do contexto (i.e., localizadas).

Com base num sistema em desenvolvimento que suporta informação turística de cariz temático, sobre pontos de interesse ou serviços, é proposta nesta dissertação uma representação de informação temporal de forma a proporcionar suporte temporal a toda a informação disponibilizada. Esta informação temporal é utilizada, por exemplo, para a validação de consultas sobre a disponibilidade temporal de serviços e pontos de interesse turístico, bem como na elaboração de percursos turísticos temáticos, pré-definidos a disponibilizar ao turista.

Keywords

location-based services, mobile devices

Abstract

Tourism plays an important role in the Portuguese economy, has shown significant growth over the years, and is, therefore, a very competitive and attractive investment sector.

New information and communication technologies enable tourism operators to offer additional services, as an answer to the growing demand for personalized services by today's tourist. Combining localization and mobile information technologies allows the development of software applications which provide tourists with context- and location-aware services.

Based on a system being developed which supports contextualized touristic information, about services or points of interest, a representation for temporal information is proposed in this dissertation. Such temporal information enables the validation of queries regarding the availability of touristic services or points of interest, as well as the set-up of pre-defined touristic routes.

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Objectivos	2
1.3	Estrutura da dissertação	2
2	Sistemas de Informação Turística	5
2.1	Turismo	5
2.1.1	Conceitos	6
2.1.2	Turismo e informação	7
2.2	Serviços baseados na localização	7
2.2.1	Conceitos	8
2.2.2	Requisitos e componentes de uma arquitectura LBS	12
2.2.3	Desenho de sistemas de computação ubíqua	17
2.3	Bases de dados temporais	18
2.3.1	Conceitos	18
2.3.2	XML temporal	22
2.4	Guias turísticos móveis	27
2.4.1	COMPASS	28
2.4.2	CRUMPET	28
2.4.3	Cruso	29
2.4.4	CityHelp	30
2.4.5	Comparação dos sistemas	30
2.5	Sumário	33
3	Modelação Conceptual e Implementação do Sistema	35
3.1	Descrição do sistema	35
3.2	Arquitectura do sistema	37
3.3	Modelo conceptual do sistema	39
3.3.1	Requisitos	39
3.3.2	Modelo de domínio do sistema	39
3.4	Desenho e implementação	40
3.4.1	Repositório de dados	40
3.4.2	Métodos de processamento dos dados	49

3.5	Sumário	50
4	Protótipo para a Elaboração de Percursos Turísticos	51
4.1	Funcionalidade esperada	51
4.2	Ferramentas utilizadas	52
4.3	Desenho do protótipo	54
4.4	Implementação do protótipo	56
4.5	Avaliação do sistema	60
4.6	Sumário	61
5	Conclusões	63
	Apêndice A Descrição dos Processos	65
	Apêndice B Casos de Utilização	67

Lista de Figuras

2.1	Representação do modelo do contexto	9
2.2	Diagrama das tecnologias	10
2.3	Componentes básicos duma <i>LBS</i>	14
2.4	Componentes da LBS e fluxo da informação	15
2.5	Diagramas das categorias de LBS	16
2.6	<i>Schema</i> XML temporal	24
2.7	Documento XML com tempo de validade	25
2.8	elemento de documento XML com tempo de validade	26
2.9	CRUMPET	29
2.10	Cruso	30
2.11	CityHelp	31
3.1	Arquitectura do sistema - o agente turístico	38
3.2	Arquitectura do sistema - o turista	38
3.3	modelo domínio	41
3.4	Categorias e tipo de POIs	42
3.5	Categoria e tipo de serviços	42
3.6	Diagrama de classes do Horário	46
3.7	interface de introdução de horários	49
4.1	Casos de utilização	55
4.2	Criar Percurso	57
4.3	Listagem de serviços disponíveis num intervalo de tempo	58
4.4	Distância de um ponto a uma recta	60

Lista de Tabelas

2.1	Relações básicas de comparação temporal de Allen	20
2.2	Predicados temporais para comparações entre intervalos de tempo	20
2.3	Tabela com tempo de transacção	21
2.4	Tabela com tempo de transacção com actualização	22
2.5	Comparação geral entre guias turísticos móveis	31
2.6	Comparação de contextos e características gerais entre guias turísticos móveis	32
3.1	Elementos dos POIs e serviços	43
3.2	Rotas.xml	44
3.3	Registos.xml	45
4.1	Comparação geral entre guias turísticos móveis e o protótipo	61
4.2	Características gerais dos guias turísticos móveis e do protótipo	62

Capítulo 1

Introdução

O Turismo tem um papel importante na economia portuguesa e mostra um crescimento significativo ao longo dos anos, sendo por isso um sector apelativo ao investimento. As novas tecnologias vieram impulsionar ainda mais este sector, que é bastante competitivo, oferecendo novos serviços e respondendo a novas exigências dos turistas.

Actualmente, as redes de telecomunicações têm uma cobertura quase total do território nacional e a utilização de dispositivos móveis, nomeadamente telemóveis e PDA's, já faz parte do nosso dia-a-dia. Esses dispositivos apresentam-se com tamanhos cada vez mais reduzidos e com capacidade de processamento cada vez maior, permitindo a integração de novas funcionalidades e de novos serviços. Os avanços tecnológicos vieram criar novos hábitos e novas necessidades nos utilizadores. Estes esperam usufruir dos serviços em qualquer altura e onde quer que se encontrem. Surgem desta forma os serviços baseados na localização.

Quando um turista se encontra em viagem, seja no seu país ou não, necessita de informação, para, por exemplo, saber onde ficará alojado, quais os locais que visitará, etc. Mesmo tendo planeado a sua viagem com antecedência, nem sempre a informação se encontra actualizada, e erros ou lapsos acontecem.

Informar o turista quando e onde este requereu determinada informação, ou avisá-lo de algum acontecimento da sua preferência, são alguns dos objectivos dos guias turísticos móveis.

1.1 Motivação

O “ciclo de vida” de um turista pode dividir-se em três fases: antes da viagem, durante a viagem e depois da viagem. Na primeira fase, o turista, planeia a sua viagem, recorrendo a informação que consegue arranjar através de livros, publicidade (por exemplo, *flyers* ou brochuras) ou experiências vividas por conhecidos, e cada vez mais através da *Internet*. Nesta fase, o turista tenta planear visitas a locais de que ouviu falar, viu em filmes, ou leu em livros, ou ainda procura por temas em que tenha interesse, tentando organizar o

tempo segundo uma agenda. Mas não é raro que a informação adquirida pelos meios atrás referidos peque em exactidão, na actualização ou na escassez dos dados, ficando incógnitas por resolver.

Durante a viagem, o turista tenta seguir o plano previamente traçado e, para isso, precisa de adquirir a informação em falta ou confirmar a que obteve. Então recorre aos meios à disposição, tais como postos de informação turística. O problema é que estes não estão abertos 24 horas por dia nem são omnipresentes. O turista pode, no entanto, tentar socorrer-se das pessoas que vai encontrando, mas nesse caso a informação que lhe é facultada pode não estar correcta. A língua também pode ser um obstáculo.

Após a viagem, a informação de que o turista necessita é a informação angariada durante a viagem, como fotos ou vídeos. Outra informação que os turistas podem prezar ter é o percurso por onde estiveram e passaram.

O trabalho apresentado nesta dissertação foi motivado pelo desenvolvimento de um guia turístico baseado no contexto e com conteúdos relativos a Portugal, acessíveis através de dispositivos móveis. A ideia é oferecer uma solução em que parceiros, como agentes ou fornecedores de serviços com interesse turístico, possam gerir os seus conteúdos e estes possam servir para orientar e informar os utilizadores, conforme os seus perfis, e ajudá-los antes, durante e após a sua viagem.

1.2 Objectivos

O objectivo deste trabalho consiste no estudo e desenvolvimento de um protótipo de um servidor de conteúdos turísticos com funcionalidades temporais. O principal objectivo consiste no desenho e implementação de um repositório de dados para a manutenção de informação turística. Este repositório incluirá: pontos de interesse turístico, como monumentos ou museus, serviços como restauração ou alojamento entre outros, eventos sazonais ou ocasionais, como férias ou festivais, e ainda percursos preparados previamente pelos operadores turísticos. Esta informação deverá ser mantida via *Web* pelos agentes e operadores turísticos que actuarão como parceiros na produção e actualização dos conteúdos, e estar permanentemente disponível para ser descarregada para os dispositivos móveis dos turistas. Os processos de transferência de dados de e para o servidor devem ser simples.

1.3 Estrutura da dissertação

- O segundo capítulo contextualiza a dissertação, discutindo a importância da informação no Turismo. Definem-se os serviços baseados na localização, e apresentam-se os requisitos e os componentes necessários à sua arquitectura. Introduzem-se os conceitos sobre as bases de dados temporais assim como o XML temporal. Este capítulo termina com a apresentação de alguns guias turísticos móveis.
- O terceiro capítulo contempla o desenvolvimento do sistema proposto. Apresentam-se os requisitos e a arquitectura do sistema implementado e das ferramentas utilizadas

na sua elaboração. Descrevem-se o desenho do sistema e os pormenores técnicos considerados no seu desenvolvimento.

- No quarto capítulo explica-se o interface do sistema, apresentam-se cenários de utilização. Para terminar discutem-se as vantagens e desvantagens deste sistema em relação aos guias turísticos existentes.
- No quinto capítulo apresentam-se as conclusões e discutem-se futuras melhorias.

Capítulo 2

Sistemas de Informação Turística

A informação tem um papel primordial para o desenvolvimento do Turismo e os turistas recorrem constantemente a esta, antes, durante e até mesmo depois das suas viagens. A *Internet* veio facilitar em muito a tarefa dos turistas, que hoje em dia já dispõem de fontes de informação vastas. No entanto, os avanços tecnológicos continuam e a massificação dos hábitos de utilização de dispositivos móveis, cada vez com maiores capacidades, proporciona o desenvolvimento de novos serviços com elevado potencial de aplicação no Turismo. Neste contexto, existem diversas dimensões da informação turística a considerar, nomeadamente, a dimensão geográfica (local a visitar); a dimensão temporal (data da visita e os eventos programados nesse período). Este capítulo começa por apresentar conceitos básicos a considerar num sistema de informação turística, e faz uma revisão sobre novos serviços baseados na localização e sobre modelos de dados para a representação de informação temporal. No final, são apresentadas algumas soluções de guias turísticos orientados para dispositivos móveis já propostas.

2.1 Turismo

Portugal registou em 2006, cerca de 11,3 milhões de chegadas de turistas, um aumento de 6,3%, posicionando-se na 19^a posição do ranking mundial dos países com mais chegadas e em 12^o lugar na Europa [1]. Segundo o World Travel & Tourism Council (WTTC¹) prevê-se que para o ano de 2008 o sector do Turismo contribua com 6,4% do PIB nacional, e estima-se que a nível de empregos represente 19% da população activa [2], sendo por isso de grande importância na economia portuguesa.

¹O Conselho Mundial de Viagens e Turismo é um fórum de líderes empresariais da indústria de Viagens e Turismo. Com directores executivos de cerca de uma centena de empresas líderes mundiais de Viagens e Turismo como membros, o WTTC tem um mandato único e visão geral sobre todos os assuntos relacionados com Viagens e Turismo (http://www.wttc.org/eng/About_WTTC/).

2.1.1 Conceitos

Segundo o WTTC as formas básicas de Turismo são:

- o **Turismo doméstico**, envolvendo residentes de um dado país a viajar unicamente dentro desse país;
- o **Turismo para o interior**, envolvendo não residentes viajando no país;
- o **Turismo para o exterior**, envolvendo residentes a viajar para outro país.

Estas formas de Turismo também se verificam quando são referentes a regiões e estas podem ser áreas dentro de um país ou um grupo de países. A partir destas três formas básicas, obtém-se as seguintes categorias de Turismo [5]:

- o **Turismo interno**, que inclui o Turismo doméstico e o Turismo para o interior;
- o **Turismo nacional**, que inclui o Turismo doméstico e o Turismo para o exterior;
- o **Turismo internacional**, que consiste no Turismo para o interior e o Turismo para o exterior.

O conceito global de “viajante” refere-se a “qualquer pessoa em viagem entre dois ou mais países ou entre duas ou mais localidades, no seu país de residência habitual. Um viajante internacional é definido como qualquer pessoa em viagem fora do seu país de residência”. Um “viajante doméstico” por sua vez é definido como qualquer pessoa em viagem dentro do seu país de residência. Estas duas definições são independentes da finalidade da viagem e dos meios de transportes utilizados. É feita ainda uma distinção entre dois grandes tipos de viajantes: os “visitantes” e os “outros viajantes”. Todos os tipos de viajantes que fazem Turismo são descritos como visitantes. Por isso o termo “visitante” representa o conceito básico de todo o sistema estatístico de Turismo. Um visitante é qualquer pessoa que viaja para outro local que o seu ambiente habitual por um período menor que 12 meses cuja finalidade da viagem seja outra que a do exercício de uma actividade remunerada no país visitado. Com o propósito estatístico e em conformidade com as formas básicas de Turismo, um visitante pode ser classificado como [5]:

1. **Visitante internacional** que descreve qualquer pessoa que viaja para outro país fora da sua residência habitual e do seu ambiente normal, por um período inferior a 12 meses e em que o principal objectivo da visita não é o exercício de uma actividade remunerada naquele país, e inclui:
 - (a) **Turistas (visitantes que pernoitam)** são os visitantes que permanecem pelo menos uma noite num alojamento colectivo ou particular no país visitado.
 - (b) **Visitantes de um dia** são visitantes que não passam a noite num alojamento colectivo ou privado no país visitado.

2. **Visitantes domésticos** são descritos como qualquer pessoa residente num país, que viaja para um local dentro de um país, fora do seu ambiente habitual por um período não superior a 12 meses e cujo principal objectivo da visita é outro que o exercício de uma actividade remunerada no local visitado. Este também distingue as categorias de turistas (visitantes que pernoitam) e visitantes de um dia.

2.1.2 Turismo e informação

A informação para um visitante é crucial e antes de partir em viagem para um determinado destino, este recorre a informação que geralmente encontrará em guias, livros e na *Internet*. Já no decorrer da viagem a necessidade de informação permanece [3].

Antes da viagem, o visitante traça, mediante os seus interesses, possibilidades e em função dos bens e serviços que poderá encontrar no destino, um percurso com locais a visitar. De modo a aproveitar ao máximo a viagem, necessita planear com antecedência tentando assim minimizar perdas de tempo ou dinheiro. Para a pesquisa de tal informação, o visitante recorre geralmente a livros, brochuras e *Internet*.

Durante a viagem, o visitante continua com necessidades de informação e, em caso de necessidade, pode recorrer a postos turísticos ou de atendimento. Aí encontrará brochuras, folhetos e também já começará a encontrar quiosques de informação. Contudo, este método tradicional de distribuição (brochuras) nem sempre é eficaz. São frequentemente desorganizadas, desactualizadas, pouco atractivas e não possuem espaço suficiente para a informação que devem conter, tornando-se difícil para o viajante encontrar a informação específica que procura e necessita [4]. Os quiosques, por sua vez, são uma boa alternativa, fáceis de utilização, com informação actualizada e boa apresentação. Mas estes não são ubíquos e são limitados ao tempo de funcionamento dos locais onde se podem encontrar.

Com os desafios que o Turismo tem que enfrentar no sentido de atingir um leque cada vez maior de consumidores heterogêneos e de diversificar os seus produtos, tendo em conta as modificações nos hábitos de vida e motivações dos consumidores, torna-se premente inovar e incrementar a competitividade e eficiência das infra-estruturas turísticas o que passará, sem dúvida, por um recurso às novas tecnologias. A informação é crucial para o sector do Turismo. Poder-se-á, mesmo, afirmar que sem ela este não sobreviveria [3].

2.2 Serviços baseados na localização

Os sistemas baseados na localização (LBS²) permitem aos utilizadores acederem a informação relativa à sua localização geográfica. Este conceito não é recente, e historicamente não foi inventado com o aparecimento das telecomunicações móveis, bastando recordar, por exemplo, os cartazes de uma peça de teatro, afixados junto do mesmo, ou até os *flyers* publicitários distribuídos de mão em mão. Outro exemplo ilustrativo são os sinais de trânsito. Também existem LBS centrados na noção do posicionamento das pessoas, umas

²do inglês, Location Based Services.

em relação às outras [9]. Mais genericamente estes serviços são também conhecidos por serviços baseados no contexto.

2.2.1 Conceitos

Definição

Na literatura científica é possível encontrar várias as definições sobre os *LBS*:

- são serviços de informação baseados na localização disponibilizados para consulta a partir de dispositivos móveis, via uma rede móvel [13].
- são serviços IP *wireless* que utilizam a informação geográfica para servir utilizadores móveis. Esta definição inclui qualquer serviço ou aplicação que tire partido da posição dos terminais móveis [11].
- um LBS numa rede de telefones móveis é um serviço fornecido a um assinante baseando-se na sua localização [12].

Contexto

Por sua vez, o conhecimento sobre o contexto dos utilizadores pode ser explorado tendo em vista proporcionar novos serviços e mais valias. Em [27] define-se o contexto como sendo "qualquer informação que pode ser utilizada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, lugar ou objecto que é considerado relevante para a interacção entre um utilizador e uma aplicação, incluindo o utilizador e a aplicação". Este conceito geral de contexto pode ser dividido noutros elementos mais específicos. Tarasewich [20] considera três grandes categorias de contexto, abrangendo o ambiente, os participantes e as actividades (figura 2.1). A categoria do ambiente respeita às propriedades dos objectos num ambiente físico. A categoria dos participantes inclui propriedades pessoais assim como expectativas do utilizador e a localização. A categoria das actividades descreve as tarefas e os objectivos do utilizador. As diferentes categorias podem ter inter-relações, por exemplo, relações participantes versus ambiente ou utilizador em casa versus utilizador no trabalho. A cronologia do tempo também é considerada, permitindo contextos históricos e previsão de eventos futuros.

As três categorias podem ainda ser divididas em diferentes factores de contexto, dos quais se passam a citar os mais importantes para a área de serviços móveis [27]:

- Localização: A localização é o principal factor de contexto. Esta informação está associada a uma determinada posição e hoje em dia existem diversas tecnologias que permitem obter esta informação com diferentes precisões.
- Tempo: Este factor pode incluir o tempo do dia ou o tempo do ano. Ter conhecimento do contexto com respeito ao tempo do dia permite, por exemplo, apresentar

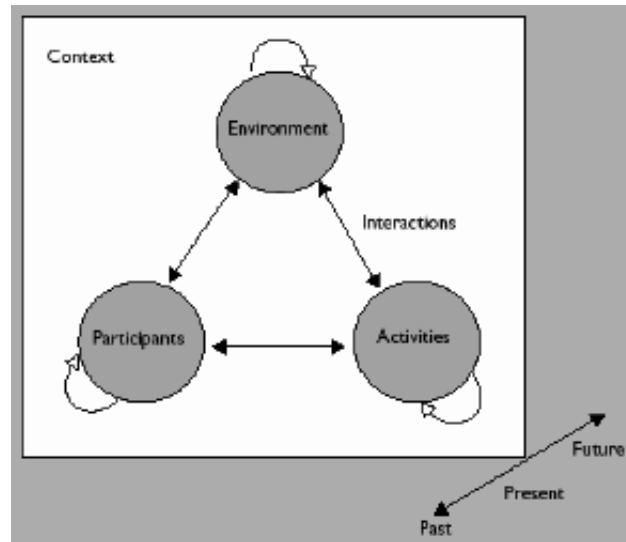


Figura 2.1: Representação do modelo do contexto [20]

as horas de abertura de um edifício aos utilizadores. O tempo do ano permite distinguir entre diferentes épocas, por exemplo, o Verão e o Inverno. Reichenbacher [21] combina os factores localização e tempo para o contexto situação. Reichenbacher entende o contexto situação no sentido de estar situado, isto é, estar num sistema de referência espaçotemporal, este não deve ser confundido com o significado de situação na definição vista em [22], onde se caracteriza todo o ambiente.

- Meio físico: o meio físico do utilizador pode ser dividido em ruído de fundo, iluminação, temperatura e clima. Por exemplo, um elevado nível de ruído de fundo pode impedir a utilização de áudio como meio de introdução ou saída de dados. O ecrã dos dispositivos móveis é de difícil leitura durante o dia, e por isso as cores da interface do utilizador e os mapas devem ser adaptados ao brilho da luz. A temperatura e as condições climáticas podem alterar o significado de proximidade do utilizador.
- Histórico de navegação: o registo dos trajectos anteriores do utilizador pode tornar-se útil para sugerir futuramente trajectos que tenham *pontos de interesse* (POIs) semelhantes, trajectos adequados ao utilizador ou para visualizar o caminho realizado pelo utilizador num mapa.
- Orientação: informação sobre a orientação do utilizador permite apresentar o mapa na posição correcta em relação à direcção dos movimentos do utilizador ou apresentar os principais POIs ou pontos de referência do campo de visão do utilizador. "Estudos mostram que os utilizadores preferem mapas com orientação egocêntrica, que são alinhados à direcção do movimento" [27]. Com o uso de mapas egocêntricos já não é necessário orientar manualmente os mapas e dessa forma reduz-se o esforço cognitivo dos utilizadores em fazer corresponder a realidade e o mapa.

- Cultural e social: Os contextos sociais incluem raciocínios sobre o utilizador, esteja ele só ou num grupo de pessoas. O contexto cultural dá informação sobre a situação do utilizador. As cores podem, consoante as áreas culturais, ter significados diferentes e a informação pode ser apresentada em linguagens diferentes.
- Sistema: as características do sistema utilizado também constituem um factor de contexto importante. Vistos os dispositivos móveis serem cada vez mais, os utilizadores esperam ser capazes de usar as mesmas aplicações nos vários dispositivos independentemente das suas capacidades ou limitações (por exemplo, tamanho e tipo do ecrã, capacidade do processador). Os dispositivos móveis necessitam de ligações a redes de comunicação, e a qualidade da conectividade fornecida nem sempre é a mesma. As características da rede terão então influência no modo como a informação é transmitida e visualizada.
- Utilizador: o contexto do utilizador inclui atributos demográficos tais como o sexo, a idade, assim como aspectos pessoais, como por exemplo as preferências, as experiências, as competências ou as histórias vividas.
- Objectivo de utilização: um dos mais importantes factores é o objectivo da utilização de aplicações de conhecimento de contexto.

Os LBS surgem da convergência dos *Sistemas de Informação Geográfica* (GIS³), da *Internet* e das novas tecnologias da informação e comunicação⁴ (figura 2.2).

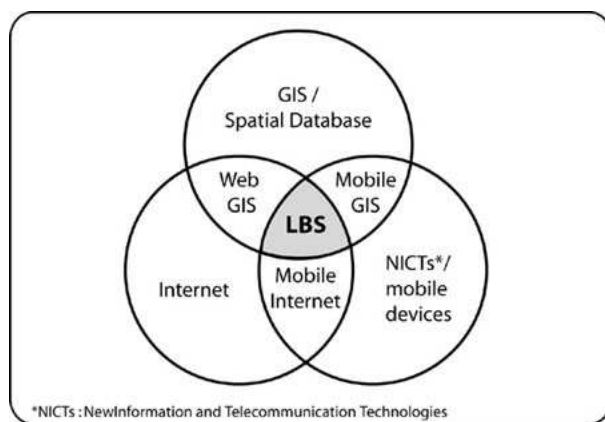


Figura 2.2: Diagrama das tecnologias [10]

Sistemas de informação geográfica

Cláudia Medeiros refere a existências de diversas definições de sistemas de informação geográfica (GIS) baseados no tipo de utilizador e no domínio da aplicação [7]. Refere ainda

³do inglês Geographic Information Services

⁴*New Information and Communication Technologies*

que um GIS deve gerir dados e realizar operações de extracção de dados georreferenciados⁵. Os GISs têm um vasto domínio de aplicações, por exemplo: planeamento urbano, gestão de redes públicas, demografia, cartografia, agricultura, gestão de recursos naturais, monitorização costeira, controlo de fogos e epidemias. Cada tipo de aplicações lida com diferentes funcionalidades, escalas e propriedades geográficas [7]. Os dados geográficos têm três características principais: a sua posição geográfica (coordenadas); os seus atributos (valores) e as suas relações topológicas. Guardada esta informação num GIS, estes dados podem ser classificados em três categorias [7]:

- *dados convencionais* – atributos tradicionais alfanuméricos que, podem ser tratados por SGBD⁶ convencionais. Estes dados também são designados por dados temáticos.
- *dados espaciais* – atributos que descrevem a geometria e localização de fenómenos geográficos. Os dados espaciais têm propriedades geométricas (forma, tamanho) e topológicas (intersecção, adjacência, entre outros).
- *dados gráficos* – atributos que guardem imagens (por exemplo fotos) e são processados por funções de processamento de imagem.

Novas tecnologias de informação e comunicação

As redes de telecomunicações móveis têm-se desenvolvido de forma muito rápida, desde a segunda geração baseada no sistema global de comunicações móveis (GSM⁷) até à terceira geração com o UMTS⁸. Nesse período verificou-se um aumento significativo da largura de banda, que passou de 9.6Kbps a 2Mbps. O desenvolvimento das telecomunicações móveis também transformou os serviços que se centravam na transmissão de voz em diversas aplicações de transmissão de informação multimédia. Dispositivos sem fios e de tamanho reduzido, tais como PDA⁹ e telefones móveis, foram melhorados e podem ligar-se via infravermelhos, modem GSM ou sinais de rádio a redes *wireless*. Com o desenvolvimento das tecnologias de localização e o crescimento do número de utilizadores, os serviços fornecedores de informação baseada na localização tornaram-se nas principais aplicações destas novas tecnologias [15].

Internet

A *Internet* é numa forma simplista uma rede mundial de computadores, que começou por ser usada em instituições militares e académicas e, a partir dos anos 90, vulgarizou-se. Hoje em dia é considerada como um instrumento de divertimento, de trabalho ou de negócio, graças à quantidade colossal de informação, na sua grande parte gratuita, e ao

⁵O termo georreferenciado refere-se dados relativos a fenómenos geográficos associados às suas localizações, espacialmente referenciadas à Terra.

⁶Sistema de gestão de bases de dados

⁷*Global System Mobile*

⁸*Universal Mobile Telecommunication System*

⁹*Personal Digital Assistant*

número de utilizadores que consegue alcançar. As operações que se consegue executar não param de crescer, por exemplo, pode falar-se do “*homebanking*”, serviços facultados pelos bancos, e permitem desde casa, executar diversas operações bancárias, como transferências, pedido de cheques ou pagamentos. O próprio Estado tem facultado diversas operações aos cidadãos como entregar a declaração de IRS, fazer o pedido de certidões. No âmbito deste projecto, um serviço interessante é o *Google Maps*, que permite visualizar mapas e fotos de satélite com um detalhe impressionante de quase todo o mundo, e permite calcular o trajecto entre dois pontos, a pé ou de automóvel, fornecendo as indicações para o realizar.

Com o aparecimento dos dispositivos móveis e a sua rápida proliferação, a *Internet* passou a ser acessível em quase toda a parte. A *Internet* mudou a forma de viver de muitas pessoas. E está a evoluir para uma interacção e participação cada vez maior dos seus utilizadores.

Serviços pull e push

Existem 2 grandes tipos de serviços de localização [13]:

Os **pull services** entregam informação directamente requisitada pelo utilizador. Este serviço pode ainda ser separado em **serviços funcionais**, como chamar um táxi ou uma ambulância, carregando apenas num botão, ou **serviços de informação** como pesquisar a farmácia mais próxima.

Push services entregam informação que pode não ser directamente requisitada pelo utilizador. Estes serviços são activados por eventos que podem ser lançados, por exemplo, quando se entra numa determinada zona ou a uma determinada hora. Como estes serviços não são lançados por intervenção directa do utilizador, são mais complexos de criar, pois requerem a existência de informação sobre o perfil, as necessidades ou as preferências dos utilizadores.

2.2.2 Requisitos e componentes de uma arquitectura LBS

As necessidades dos utilizadores têm tendência a evoluir com os desenvolvimentos tecnológicos e os sistemas devem acompanhar essa evolução, obrigando a novos requisitos nas arquitecturas dos sistemas LBS. Por exemplo, nos sistemas de informação geográfica, normalmente existe um servidor de aplicações que irá servir um número limitado de utilizadores que realizam tarefas relativamente complexas. Por outro lado, os LBS oferecem acesso e informação com elevado número de utilizadores com um grau de complexidade inferior [14].

Desta forma existe um conjunto de requisitos específicos para uma boa implementação dos sistemas LBS:

- **Alto desempenho:** entregar respostas em fracções de segundos a um número elevado de pedidos que podem ocorrer em simultâneo.

- **Arquitectura escalável:** suportar milhares de utilizadores em paralelo e grandes quantidades de informação.
- **Fiabilidade:** dar respostas sem erros.
- **Concorrência:** suportar a gestão de informação dinâmica, actualizada por vários utilizadores em simultâneo, de forma consistente.
- **Mobilidade:** disponível em todos os dispositivos e em qualquer localização.
- **Aberto:** suportar normas e protocolos comuns (HTTP¹⁰, WAP¹¹, WML¹², XML¹³, MML¹⁴).
- **Segurança:** gerir serviços de segurança capazes de garantir a protecção e a confidencialidade dos dados.
- **Interoperabilidade:** integrado com aplicações *e-Business*, tais como gestão das relações com os clientes, facturação, *gateways* de posicionamento *wireless*.

Para contemplar todos estes requisitos, que tornam a arquitectura dos LBS complexa, são necessários acordos entre as diversas entidades, como fornecedores de *hardware* e de *software*, fornecedores de serviços *online* e de conteúdos, fornecedores de redes *wireless* e de infra-estruturas, vendedores de equipamento *wireless* e portais da Internet. Só com acordos entre estes parceiros se consegue proporcionar uma oferta satisfatória para os utilizadores e uma utilização eficaz dos serviços.

Um *LBS* envolve os seguintes componentes [8]:

- **Dispositivos móveis:** dispositivos usados para pedir informação. Os resultados podem ser apresentados como áudio, vídeo, imagem, texto, etc. Estes dispositivos podem ser portáteis, como *PDA*s, ou telemóveis, entre outros.
- **Rede de comunicação:** esta componente é a rede móvel responsável por encaminhar os pedidos dos dispositivos móveis até ao fornecedor de serviços *Internet*.
- **Componente de posicionamento:** esta componente é responsável pela determinação da posição do utilizador. A sua localização pode ser determinada pela rede móvel, através de triangulação; os sistemas de posicionamento Global (GPS); as redes sem fios (WLAN); antenas de rádio, sendo que estas últimas têm vantagens em locais fechados como por exemplo os museus. Em último caso a posição pode ser especificada manualmente pelo utilizador.

¹⁰ *Hypertext Transfer Protocol* - protocolo de comunicação para transferir informação na *Internet*

¹¹ *Wireless Application Protocol* - protocolo de acesso à *Internet* a partir de dispositivos móveis

¹² *Wireless Markup Language* - linguagem concebida para o WAP, para mostrar conteúdos nos ecrã dos telemóveis

¹³ *eXtensible Markup Language*

¹⁴ *Multimedia Markup Language*

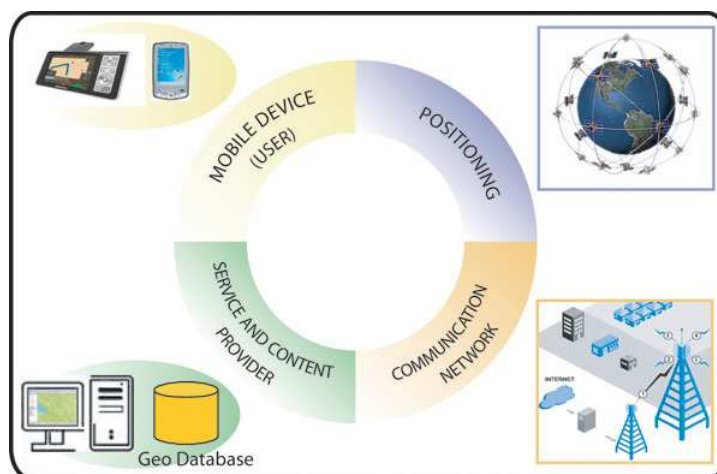


Figura 2.3: Componentes básicos duma *LBS* [8]

- **Fornecedores de serviços e aplicações:** Estes fornecedores oferecem diversos serviços aos utilizadores e são responsáveis pelo processamento dos pedidos que são feitos pelos assinantes. Os serviços oferecidos podem ser o cálculo de posições, encontrar um percurso, procurar nas páginas amarelas através da posição ou procurar informação específica sobre objectos, entre outros.
- **Fornecedores de informação e conteúdos:** Em geral, os fornecedores não recolhem nem preservam toda a informação que possa vir a ser requisitada pelos utilizadores. Por exemplo, os dados geográficos e a informação sobre os locais são geralmente solicitados às autoridades competentes, parceiros de negócio ou industriais (por exemplo, páginas amarelas ou companhias de tráfego).

Funcionamento e alguns exemplos de implementações

De seguida, passa-se a explicar qual o trajecto e como é processado um pedido numa *LBS* (figura 2.4):

1. Aquando da execução do pedido, a posição actualizada do dispositivo móvel é obtida pelo sistema de posicionamento. A localização pode ser obtida tanto pelo dispositivo, através do *GPS*, como também pode ser obtida por um serviço de posicionamento da própria rede. Depois o cliente envia o pedido, que contém o objectivo da procura e a posição actual para o *Gateway* ¹⁵.
2. O *Gateway* tem como tarefa trocar mensagens entre a rede de comunicação e a *Internet*. Portanto sabe os endereços *Web* de vários servidores de aplicações e encaminha o pedido para um desses servidores. O *Gateway* também guardará a informação relativa ao dispositivo móvel que fez o pedido.

¹⁵Nó numa rede que serve de porta de entrada para outra rede e vice-versa

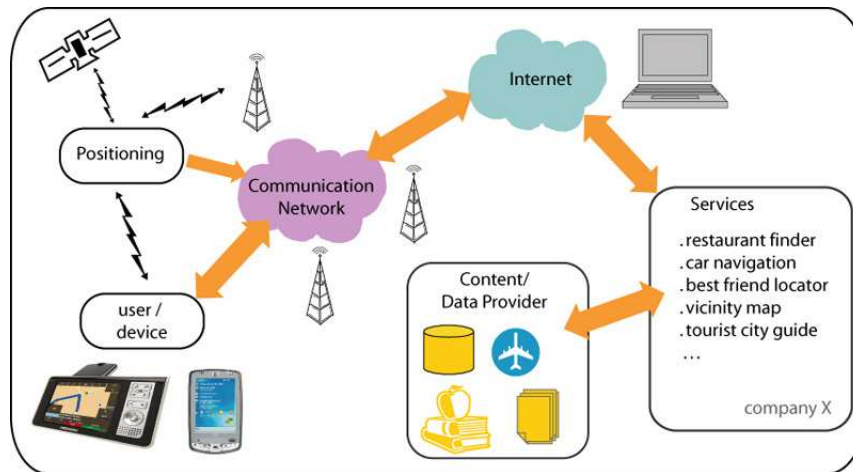


Figura 2.4: Componentes da LBS e fluxo da informação [8]

3. O servidor de aplicações lê o pedido e activa o respectivo serviço.
4. O serviço analisa a mensagem e determina a informação adicional que, para além dos critérios de pesquisa e da posição do utilizador, é necessária para responder ao pedido.
5. Depois, o serviço processará a informação comunicando com os respectivos fornecedores de dados ou conteúdos. Por exemplo, na procura de um percurso verifica se existe caminho entre a origem e o destino.
6. Estando toda a informação está pronta para ser enviada ao utilizador, o serviço reencaminhará a resposta ao pedido via *Internet*, *gateway* e pela rede de comunicação.

Por fim o resultado da pesquisa é apresentado no dispositivo móvel, de acordo com os critérios e nos formatos solicitados. O utilizador poderá fazer novos pedidos ou refinar a pesquisa anterior.

Existem diversas aplicações dos LBS. A figura 2.5, dá uma visão geral das principais categorias e implementações dos LBS.

1. Serviços de emergência

Uma das aplicações mais evidentes para as LBS é a capacidade de localizar um indivíduo que desconhece a sua localização exacta ou não é capaz de a transmitir. Outro exemplo, quando um condutor desconhece a sua localização e necessita de assistência. Nestes dois casos concretos, com a localização exacta transmitida automaticamente para os serviços de emergência, a assistência pode ser prestada rapidamente e de modo eficiente. Esta categoria inclui serviços de emergência, tanto públicos como privados para pessoas como para viaturas.

2. Serviços de navegação

A necessidade de um utilizador móvel conhecer a direcção a seguir do local em que

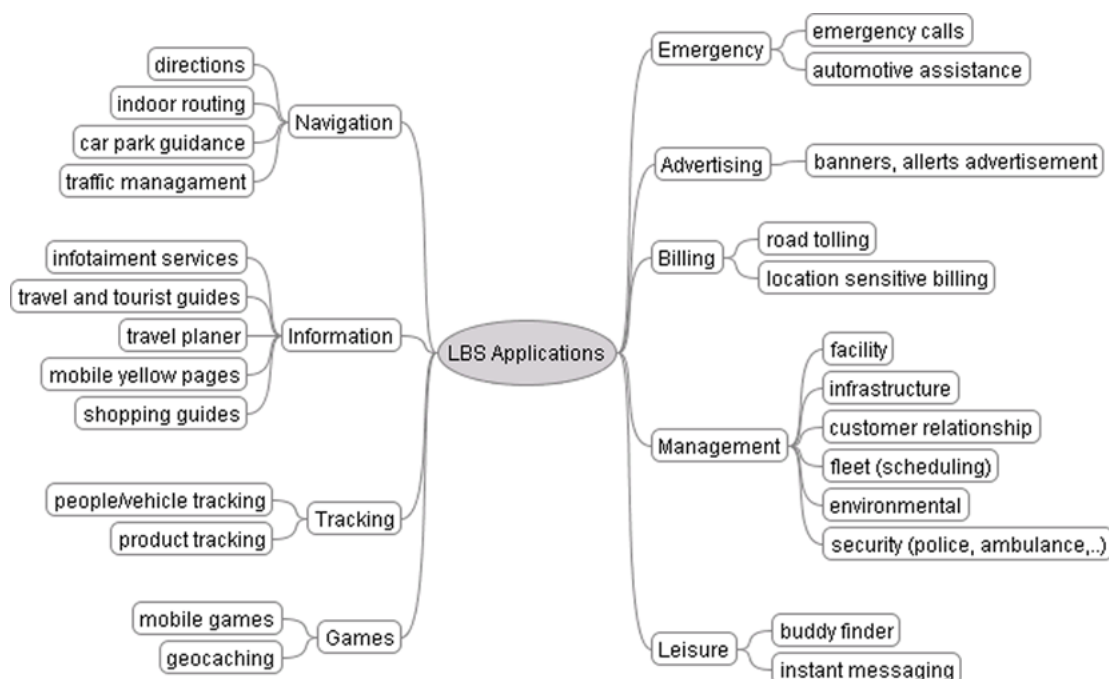


Figura 2.5: Diagramas das categorias de LBS [8]

se encontra até ao seu destino. A capacidade que o operador tem de localizar o assinante, de calcular o percurso e indicar-lhe o caminho.

3. Serviços de informação

Encontrar o serviço mais próximo, obter informação sobre o trânsito, obter informação turística sobre um monumento, obter um mapa de uma vila ou cidade desconhecida, estes são alguns dos muitos exemplos possíveis de serviços de informação.

4. Serviços de gestão e de rastreio

Este serviço pode ser útil tanto para os clientes como para as empresas. Por exemplo, conhecer o trajecto de uma encomenda postal, saber onde se encontram as viaturas de empresas de transporte, podendo dessa maneira enviar a viatura mais próxima ao novo pedido, ou no caso de uma empresa de transportes públicos informar os clientes de possíveis atrasos.

5. Serviço de cobrança

A cobrança sensível à localização refere-se à capacidade que um fornecedor de serviços móveis de localização tem para se pagar automaticamente de um utilizador que tire partido de um determinado serviço dependendo da sua localização, ou acedendo a determinado serviço.

2.2.3 Desenho de sistemas de computação ubíqua

Khai Truong [16] coloca algumas questões que se levantam quando se projectam aplicações de captura e de acesso a informação na computação ubíqua. Essas questões também são colocadas na concepção dos guias turísticos pessoais baseados no contexto. Em analogia ao trabalho de Khai Truong, identificam-se as questões levantadas:

- **Quem?** Determinar quem são os utilizadores, quantos são e quais as funções que desempenham no sistema. O perfil dos utilizadores também é importante.
 - Quem introduz a informação na base de dados?
 - Quem acede a informação?
 - Qual o número de utilizadores?
- **O quê?** Quais os conteúdos a armazenar na base de dados e quais os conteúdos a disponibilizar aos intervenientes?
 - Que informação guardar?
 - Que informação disponibilizar?
- **Quando?** Focar aspectos temporais:
 - Quando ocorrem os pedidos?
 - Quando irá realizar-se o trajecto?
 - Quando está aberto o museu?
- **Onde?** Questões relativas à localização onde é feito o pedido e à localização do objectivo do pedido. Saber onde ocorrem os pedidos permite responder melhor aos pedidos feitos.
 - Onde ocorrem os pedidos?
 - Onde encontrar este estabelecimento?
 - Onde ocorrem sobreposição dos espaços físicos?
 - A mobilidade dos utilizadores.
 - A multiplicidade dos locais.
- **Como?** O tipo e número de dispositivos que são utilizados no sistema e a escalabilidade dos dispositivos.
 - Qual a função dos dispositivos móveis?
 - Como serão feitos os pedidos?
 - Como será apresentada a informação?

2.3 Bases de dados temporais

A implementação de modelos de dados não temporais como Sistemas de Gestão de Base de Dados (SGBD) captura um único estado do mundo, geralmente o mais recente. Estes SGBD suportam operações de modificação que facilitam a transição de um estado para outro, substituindo os valores existentes por novos. Nas consultas, é assumido que os dados são síncronos, isto é, que todos os factos armazenados na base de dados (BD) são válidos no momento em que a consulta é avaliada. Existem, no entanto, diversos domínios de aplicações onde é necessário guardar em paralelo estados anteriores e até estados futuros (por exemplo projectos, orçamentos, sistemas de reservas). Embora não seja um grande problema arranjar maneira de adicionar períodos de tempo de validade aos dados de forma a guardar um histórico, a dificuldade reside na pesquisa de dados que variam no tempo e especificar restrições de integridade que se mantêm nos vários estados da BD.

2.3.1 Conceitos

Modelos de tempo

A comunidade das Bases de Dados Temporais (BDT) surgiu com três modelos básicos de tempo [18]: o modelo contínuo, o modelo denso e o modelo discreto. O tempo é por natureza contínuo e o primeiro modelo aproxima-se dessa visão. Assim, o tempo pode ser representado num eixo de coordenadas de reais, e cada ponto corresponde a um instante no tempo. Esta abordagem modela o tempo com maior precisão. No modelo denso, o tempo é isomorfo aos números racionais. O modelo discreto, por sua vez, mapeia o tempo num eixo de coordenadas inteiras. Como os modelos temporais são utilizados por computadores digitais, não é possível mapear o primeiro modelo sem perda de precisão.

Granularidade do tempo

Como referido atrás, o tempo pode ser mapeado segundo um eixo coordenado. O eixo do tempo pode ser visto como um conjunto de pontos. A unidade de tempo mais pequena e indivisível considerada designa-se *chronon*. Um *chronon* é um sub-intervalo de duração fixa no eixo do tempo. Uma aplicação pode considerar diferentes granularidades (minutos, horas, dias) permitindo desta forma aproximar o modelo ao mundo real.

Instante de tempo e eventos

Um instante de tempo representa um ponto no eixo do tempo, enquanto que um evento é um facto instantâneo, por exemplo, algo que ocorre num instante de tempo. Como o *chronon* é a unidade mais pequena representada no eixo do tempo, os modelos de tempo têm de ser mapeados em *chronons*. Então diz-se que um evento ocorre no *chronon* t se ocorrer em qualquer altura durante esse *chronon* [18].

Intervalo de tempo e elementos de tempo

Um intervalo de tempo representa um período de tempo que começa no instante de tempo S e termina no instante E , sendo estes instantes os limites inferior e superior do respectivo intervalo. Estes limites inferior e superior podem ser comparados recorrendo aos predicados $<$ e \leq . As possibilidades de especificar um intervalo de tempo são:

$$[S - E] = \{t | S \leq t \leq E\}$$

$$[S - E) = \{t | S \leq t < E\}$$

$$(S - E] = \{t | S < t \leq E\}$$

$$(S - E) = \{t | S < t < E\}$$

onde t é um instante de tempo.

Como se pode ver, um intervalo pode ser aberto, se os seus limites não pertencem ao intervalo, semi-aberto, se um dos seus limites pertencer, e fechado quando ambos os seus limites pertencerem ao intervalo. Para os intervalos estão definidas as operações união, intersecção e diferença dos conjuntos. A intersecção de conjuntos é utilizada, por exemplo, para determinar quando dois factos ocorrem em simultâneo. A união permite determinar o período global de tempo de validade de vários factos. A diferença de dois intervalos permite determinar o período de tempo no qual um facto é verdadeiro e outro não. Contudo estas operações não são fechadas no que respeita a intervalos. Da união de dois intervalos disjuntos resulta um conjunto de intervalos. A diferença de dois intervalos pode resultar em zero, um ou dois intervalos.

Exemplo 2.1 *Sejam os intervalos $I_1 = [1980 - 1990)$, $I_2 = [1992 - 1994)$ e $I_3 = [1975 - 1996)$.*

Da união de I_1 com I_2 resulta:

$$I_1 \cup I_2 = \{[1980 - 1990), [1992 - 1994)\}$$

A diferença dos intervalos de tempo I_1 com I_3 resulta no intervalo de tempo vazio:

$$I_1 - I_3 = \{\}$$

enquanto que a diferença dos intervalos I_3 e I_2 devolve um conjunto de intervalos:

$$I_3 - I_2 = \{[1975 - 1992), [1994 - 1996)\}$$

Com o exemplo 2.1 [18], introduz-se a noção de domínios temporais que são uniões finitas de intervalos de tempo. Este conjunto de intervalos também se designam por elementos temporais. Os resultados das operações união e diferença do exemplo 2.1 são *elementos temporais*.

Um elemento de tempo é um conjunto finito de intervalos disjuntos ou não. Os elementos de tempo também podem conter instantes de tempo. Como se pode ver no exemplo 2.1, operações de conjuntos sobre intervalos de tempo podem gerar elementos de tempo. E todas as operações realizadas sobre elementos de tempo, i. e., a união, a intersecção, a diferença e o complemento produzem elementos de tempo. Os elementos de tempo são por isso, fechados para as operações sobre conjuntos. Os elementos permitem representar situações que no caso dos intervalos levariam a duplicar informação na BD.

Predicados de comparação dos intervalos de tempo

Allen [19] introduziu um conjunto de treze predicados de comparação temporal para intervalos de tempo. Estes descrevem todas as relações possíveis entre dois intervalos. Na tabela 2.1 apresentam-se esses predicados assim com a sua definição. Steiner [18] refere outro conjunto de predicados, válido para instantes de tempo (ver tabela 2.2).

Comparação	limites
x antes de y	$x^+ < y^-$
x depois de y	$y^+ < x^-$
x encontra y	$x^+ = y^-$
x encontrado por y	$y^+ = x^-$
x sobrepõe y	$x^- < y^- < x^+ \wedge x^+ < y^+$
y sobreposto por x	$y^- < x^- < y^+ \vee y^+ < x^+$
x durante y	$x^- > y^- \wedge x^+ < y^+$
y inclui x	$y^- < x^- \wedge y^+ > x^+$
x inicia y	$x^- = y^- \wedge x^+ < y^+$
x iniciado por y	$x^- = y^- \wedge y^+ < x^+$
x acaba y	$x^- > y^- \wedge x^+ = y^+$
x acabado por y	$y^- > x^- \wedge x^+ = y^+$
x igual y	$x^- = y^- \wedge x^+ = y^+$

Tabela 2.1: As treze relações básicas de comparação temporal de Allen [19]

Na tabela 2.1, omitiram-se as relações $x^- < x^+$ e $y^- < y^+$ que são válidas para todas as relações.

Predicados de comparação	Predicado equivalente para instantes de tempo
I_1 precede I_2	$\text{end}(I_1) < \text{begin}(I_2)$
I_1 sobrepõe I_2	$\exists t : t \in I_1 \wedge t \in I_2$
I_1 encontra I_2	$\text{end}(I_1) = \text{begin}(I_2)$
I_1 contém I_2	$\text{begin}(I_1) \leq \text{begin}(I_2) \wedge \text{end}(I_1) \geq \text{end}(I_2)$
$I_1 = I_2$	$\text{begin}(I_1) = \text{begin}(I_2) \wedge \text{end}(I_1) = \text{end}(I_2)$

Tabela 2.2: Predicados temporais para comparações entre intervalos de tempo [18]

Noções de tempo

Existem diversos aspectos a ter em atenção numa relação entre o tempo e os dados. De forma a representá-los, foram propostos diferentes eixos de tempo que distinguem diferentes formas de modelar dados temporais [18].

Tempo de validade Uma abordagem quando é proposto um modelo de dados temporal é ser capaz de identificar quando um facto é verdadeiro no mundo real. Por exemplo, identificar o período em que um operário trabalhou numa empresa. Eventos futuros também podem ser guardados, por exemplo, quando se marca uma consulta médica, o utente saberá quando terá de se dirigir ao consultório a fim de ser examinado. A esta noção de tempo designa-se por *tempo de validade* e permite determinar quando um facto foi, é ou será verdadeiro. Um intervalo de tempo de validade é portanto o período em que o facto é verdadeiro. Esta noção de tempo é interpretada por SGBDs que suportam tempo validade, por exemplo, durante a avaliação de uma consulta ou na verificação de restrições de integridade. O tempo de validade tem de ser fornecido pelo utilizador quando adiciona ou altera os dados.

Tempo de transacção Outra abordagem na modelação de dados temporais é poder identificar a altura em que foram efectuadas operações sobre os dados da BD, guardando a evolução da informação. Desta forma, é possível guardar um histórico e identificar ou corrigir erros que surjam. A esta noção de tempo designa-se *tempo de transacção*. Os SGBDs com suporte de tempo de transacção ficam responsáveis pela actualização deste atributo. No exemplo 2.2, as tabelas 2.3 e 2.4 mostram as operações efectuadas pelo SGBD quando é actualizado o campo idade de um tuplo. Sempre que é efectuada uma operação, por exemplo a alteração do valor de um atributo (neste caso, a idade), o sistema cria uma réplica desse tuplo¹⁶; ao tuplo inicial actualiza o intervalo de tempo de transacção com término (campo T_{fim}) no momento actual; ao tuplo criado coloca o valor actualizado e inicia o intervalo do tempo de transacção (campo T_{inicio}) com o valor do momento actual. Estas operações são realizadas automaticamente pelo sistema.

Exemplo 2.2 *Actualização de um atributo quando tabela suporta tempo de transacção*

Nome	Idade	Data Nascimento	T_{inicio}	T_{fim}
Rui Araújo	17	1991-01-14	2008-01-14	»
Luís Silva	31	1976-07-03	2007-07-10	»
Maria Alves	20	1990-11-02	2007-11-24	»

Tabela 2.3: Tabela com tempo de transacção

¹⁶Um tuplo representa uma linha da tabela

Nome	Idade	Data Nascimento	T_{inicio}	T_{fim}
Rui Araújo	17	1991-01-14	2008-01-14	»
Luís Silva	31	1976-07-03	2007-07-10	2008-07-04
Maria Alves	20	1990-11-02	2007-11-24	»
Luís Silva	32	1976-07-03	2008-07-04	»

Tabela 2.4: Tabela com tempo de transacção com actualização

O valor do tempo de transacção nunca poderá ser ulterior ao momento actual visto que o tempo de transacção representa o momento em que a operação é executada. Também não faz sentido actualizar o tempo de transacção, dado que uma operação já executada não pode ser anulada. A única maneira de corrigir uma operação será de inverter a mesma, o que, acontece numa altura posterior e que por isso conduz a outro registo na BD.

Tempo definido pelo utilizador Um rótulo de tempo¹⁷ que não seja interpretado pelo SGBD, como por exemplo, a data de nascimento (ver tabelas 2.3 e 2.4) é chamado *tempo definido pelo utilizador*, porque é o próprio utilizador que interpreta a informação temporal enquanto que o SGBD trata esses dados como qualquer outro atributo [18]. Modelos de dados com tempo definido pelo utilizador fornecem, por exemplo, um tipo data ou rótulo de tempo como valores para o atributo, podendo esses valores referenciar qualquer instante no tempo, isto é passado, presente ou futuro, sendo estes fornecidos pelo próprio utilizador. A distinção entre atributos de tempo poderem ser interpretados ou não surge porque os modelos de dados temporais propostos utilizam estruturas de dados subjacentes a modelos de dados não temporais para guardar a informação, estendendo o *schema* com atributos temporais especiais. Uma consulta é avaliada utilizando uma semântica temporal, as operações algébricas acedem aos atributos especiais de tempo e então interpretam-nos. Os atributos definidos pelo utilizador nunca serão acedidos desta forma [18].

2.3.2 XML temporal

O XML¹⁸ é o acrónimo de Extensible Markup Language, linguagem de marcação como é o caso do HTML, desenvolvida pelo W3C¹⁹. É uma linguagem simples e em formato texto muito flexível derivado do SGML²⁰. Originalmente concebida para responder aos desafios da edição electrónica, o XML também tem um papel cada vez mais importante na troca duma grande variedade de dados na *Internet*. O XML foi projectado para uma fácil implementação e interoperabilidade entre o SGML e o HTML [23].

¹⁷ *timestamp*

¹⁸ <http://www.w3.org/XML/>

¹⁹ *World Wide Web Consortium* – Consórcio internacional de organizações com o objectivo de obter o máximo potencial do *World Wide Web* através do desenvolvimento de protocolos e directrizes que garantam um crescimento a longo prazo da *Web*.

²⁰ *Standard Generalized Markup Language*

Vantagens

- O XML é simples, e baseado em linguagem texto com marcadores como `<nome>` ou `<hora>`, e pode de forma clara especificar que tipo informação está delimitada por esses marcadores. Como os marcadores e os conteúdos estão juntos, a informação pode ser entendida sem a necessidade da aplicação que a criou.
- Os documentos XML são de leitura fácil pelas pessoas e podem ser facilmente processados pelos computadores. Existem muitos *parsers* e processadores para a manipulação de dados XML.
- O XML é semanticamente estruturado. A estrutura do XML também representa a semântica dos dados, que representa uma grande oportunidade para extracção de informação. Dado que documentos XML podem ser condicionados pelos seus *schemas*, a interpretação dos dados pode ser mais precisa e as palavras ambíguas nas pesquisas podem ser distinguidas pelo conteúdo onde aparecem. Uma vez que os documentos XML incluem informação sobre a estrutura dos dados, é possível extrair facilmente partes relevantes de um documento.

Fabio Grandi introduziu em [24] uma extensão temporal da *World Wide Web* baseada numa infra-estrutura XML/XSL completa para suportar tempo de validade permitindo suportar a gestão de informação histórica recorrendo a marcadores básicos e técnicas de selecção temporal introduzidas na teoria de BDT. A técnica proposta possibilita a definição explícita de informação temporal em documentos HTML/XML, e não necessita alterações nas tecnologias *Web* visto ser baseado em XML e em normas relacionadas.

Para adicionar o tempo de validade nos documentos *Web*, Fabio Grandi propõe uma extensão do XML com marcação temporal. As funcionalidades das novas etiquetas são completamente especificadas por meio de um *schema* XML adequado sem alteração do navegador com suporte XML. A proposta consiste na adição de novas etiquetas XML, `<valid>`, que servirão a definir um *contexto de validade*. O contexto de validade é usado para rotular temporalmente uma parte de um documento, ao qual é atribuído um tempo específico que será utilizado para manipulação de documentos seleccionados temporalmente. O conteúdo compreendido pelo contexto de validade pode ser de qualquer tipo, nomeadamente texto, elementos gráficos, e qualquer outra estrutura XML incluindo elementos `<valid>` encaixados. Os marcadores de tempo podem ser definidos no contexto de validade por meio de etiquetas `<validity>`, que permitem definir um intervalo temporal através dos seus limites, isto é, os valores dos atributos *from* e *to* do elemento XML `<validity>`. É possível utilizar marcadores de intervalos múltiplos, e neste caso, os marcadores temporais são elementos temporais.

No exemplo 2.3 o conteúdo do contexto de validade terá como validade a reunião dos intervalos especificados pelos nos elementos *validity* e nesse caso será o elemento temporal $[1980, 1985] \cup [1995, 2000]$

O *Schema* XML do código do exemplo 2.3, é apresentado na figura 2.6.

Este *schema* XML define os atributos *from* e *to* do elemento *validity* e usa tal como o XML temporal as datas definidas por *date*.

Exemplo 2.3 [24]

```
<valid> <!-- definition of a validity context -->
  <validity from="1980-01-01" to="1985-12-31" />
  <validity from="1995-01-01" to="2000-12-31" />

  This is text <b>valid from 1980 to 1985</b>
  but also <b>valid from 1995 to 2000</b>...
</valid>
```

```
<?xml version="1.0"?>
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
  xmlns:dt="urn:schemas-microsoft-com:datatypes">

  <AttributeType name="from" required="yes" dt:type="date" />
  <AttributeType name="to" required="yes" dt:type="date" />

  <ElementType name="validity">
    <attribute type="from" minOccurs="1" maxOccurs="1" />
  <attribute type="to" minOccurs="1" maxOccurs="1" />
  </ElementType>
  <ElementType name="valid" content="mixed">
    <element type="validity" minOccurs="1" maxOccurs="*" />
  </elementType>
</schema>
```

Figura 2.6: *Schema* XML – ValidSchema.xml [24]

```

<employees vstart="1995-01-01" vend="now">
  <employee vstart="1995-01-01" vend="now">
    <name="1995-01-01" vend="now">Bob</name>
    <title vstart="1995-01-01" vend="1997-12-31">Engineer</title>
    <title vstart="1998-01-01" vend="now">Sr Engineer</title>
    <dept vstart="1995-01-01" vend="now">RD</dept>
    <salary vstart="1995-01-01" vend="1997-12-31">65000</salary>
    <salary vstart="1998-01-01" vend="1999-12-31">70000</salary>
    <salary vstart="2000-01-01" vend="now">85000</salary>
  </employee>
</employees>

```

Figura 2.7: employees.xml - documento XML com tempo de validade [25]

XQuery

A XQuery é uma poderosa linguagem que permite consultas a documentos XML e é definida pelo W3C²¹. Fusheng Wang apresenta em [25] um exemplo com consultas temporais a documentos XML com tempo de validade.

Na figura 2.7 o documento XML representa o historial de um empregado numa empresa. Em concordância, com o exemplo de Grandi os elementos são marcados com etiquetas temporais, neste caso com *vstart* e *vend* que representam o início e o fim do intervalo em que esse elemento é válido. Note-se que *vend* pode tomar o valor “now” internamente representado por “9999-12-31” e denota o momento actual. De referir que a entidade “Bob” tem um tempo de validade igual ou superior à dos seus filhos, isto acontece porque existe uma restrição em tempo de validade em que o tempo de validade de um nó pai cobre sempre o dos seus nós filhos [25]. Wang representou o código XML em 2.7 na tabela da figura 2.8.

Com este exemplo, Wang mostra a implementação das seguintes consultas temporais em linguagem de interrogação XQuery:

- Consulta de projecção temporal que devolve a história do empregado “Bob”

```

for $t in doc("employees.xml")/employees/employee[name="Bob"]/title
return $t

```

- Consultas temporal instantâneas que devolve todos os gestores a 1994-05-06:

```

for $m in doc("depts.xml")/depts/dept
/mgrno[vstart(.)<="1994-05-06" and vend(.>="1994-05-06")]
return $m

```

²¹ver <http://www.w3.org/TR/xquery/>

Name	Title	Dept	Salary
Bob	1995-01-01 Engineer 1997-12-31	RD	1995-01-01 65000 1997-12-31
	1998-01-01 Sr Engineer		1998-01-01 70000 1999-12-31
			2000-01-01 85000
	now	now	now

Figura 2.8: Elemento de documento XML com tempo de validade [25]

- “*Slicing*” temporal onde encontra os empregados que tenham trabalhado entre 1994-05-06 e 1995-05-06

```

for $e in doc("employees.xml/employees")/employees
  /employee[toverlaps(., telement(xs:date("1994-05-06"),
    xs:date("1995-05-06") ) ) ]
return $e/name

```

A função *toverlaps(\$a, \$b)* devolve “*verdadeiro*” se um nó é sobreposto temporalmente ao outro e “*falso*” em caso contrário. A função *telement(\$a, \$b)* constrói um elemento *a* com o atributo *b*.

- *Join* temporal: encontrar a história dos empregados que cada gestor gere:

```

element manages{
for $d in doc("depts.xml")/depts/dept
for $m in $d/mgrno
return
element manage {$d/deptno, $m,
  element employees.xml {
    for $e in doc("employees.xml")/employees/employee
    where $e/deptno = $d/deptno and not (empty(overlapinterval($e,$m)))
    return ($e/name, overlapinterval($e,$m)) }}
}

```

A função *overlapinterval(\$a, \$b)* verifica se os intervalos *a* e *b* estão sobrepostos devolvendo um valor booleano.

- Agregado temporal devolve a história da média dos salários:

```
let $s := document("employees.xml")/employees/employee/salary
return avg($s)
```

- Re-estruturação temporal, encontra o período máximo de tempo contínuo durante o qual o “Bob” trabalhou sem mudar de departamento:

```
for $e in doc("employees.xml")/employees/employee[name="Bob"]
let $d := $e/dept
let $t := $e/title
let $overlaps := restructure($d, $t)
return max($overlaps)
```

A função *restructure* recebe duas listas e devolve todos os intervalos em que se sobrepõem.

- A enquanto B, procura um empregado que tenha sido contratado e tenha trabalhado no departamento “RD” enquanto “Bob” era gestor do departamento.

```
for $e in doc("employees.xml")/employees/employee
for $b in doc("employees.xml")/employees/employee[name="Bob"]
let $t := $b/title[.="manager"]
let $bd := $b/dept[.="RD"]
let $d := $e/dept [1][.="RD"]
where vmeets($d, $t) and vcontains($bd, $t)
return <employee>{$e/name}</employee>
```

A função *vmeets*(\$a,\$b) verifica se a lista *a* encontra a lista *b* devolvendo um valor booleano correspondente. A função *vcontains*(\$a,\$b) verifica se a lista *a* contém a lista *b*.

As funções auxiliares que se vão encontrando nos exemplos anteriores implementam as relações básicas de comparação temporal (tabela 2.1) e os predicados de comparação temporal de Allen (tabela 2.2).

2.4 Guias turísticos móveis

Os guias turísticos baseados no contexto já não são novidade e existem soluções que os implementam, apresentando-se a seguir três soluções com abordagens diferentes.

2.4.1 COMPASS

O “*Context-aware Mobile Personal Assistant*” oferece serviços e informações relativas ao contexto baseados na localização dos turistas quando estes estão em viagem. A localização é obtida automaticamente, tanto por GPS como através das redes móveis. O sistema oferece aos turistas mapas com objectos, que lhes possibilitam serviços adicionais, tais como reservar uma mesa num restaurante ou contactar amigos. Esta aplicação é desenvolvida numa plataforma que suporta aplicações *context-aware*²² baseadas em *webservices*²³, que operam em redes 3G e requerem ligações permanentes à rede móvel.

O sistema extrai e fornece informação sobre o contexto do utilizador contactando os serviços de contexto, como por exemplo, um GIS. Para seleccionar serviços relevantes próximos do utilizador, o primeiro critério é a localização. O contexto do utilizador inclui um perfil fornecido manualmente, que é descrito utilizando uma extensão de especificação da plataforma P3P²⁴, indicando o perfil ou os interesses do utilizador que mais tarde são actualizados automaticamente por um sistema baseado nos *feedback* do utilizador para POIs específicos. O sistema também permite integrar funcionalidades como consulta de informação sobre o clima ou o tráfego automóvel.

O contexto temporal é considerado da forma seguinte: se um utilizador tiver estado recentemente num POI, menor será a relevância prevista para aquele utilizador querer visitar um POI daquela classe. No contexto lógico, o sistema pode inferir através da velocidade de deslocamento do utilizador ou de propriedades geográficas, se o utilizador está a caminhar ou a viajar de automóvel. O utilizador pode ainda subscrever *pull* e *push services* sendo que o sistema alertará o utilizador logo que se altere o contexto.

2.4.2 CRUMPET

O projecto CRUMPET “*Creation of User-friendly Mobile Services Personalized for Tourism*” [29] foi financiado pela UE. O sistema apoia utilizadores durante as suas viagens, com informação e sugestões sobre atracções turísticas, restaurantes ou hotéis. O sistema fornece informação sobre os interesses do utilizador quando este se encontra próximo desses POIs. A posição do utilizador é obtida através do sistema GPS. A arquitectura prevê uma abordagem de personalização externa podendo integrar diversos fornecedores de serviços e conteúdos. O contexto lógico do utilizador é obtido dinamicamente, guardando um histórico das suas interacções com o sistema. A informação sobre o contexto é obtida por *pull* e *push services*.

Os conteúdos foram adaptados de modo a beneficiar os turistas com informação de acordo com as preferências dos mesmos e para a localização actual. Por exemplo, um turista quando chega a uma cidade que não conhece e dispõe de um dispositivo móvel, consegue identificar a sua posição actual e recolher informação (texto ou imagem) sobre o panorama do local actual. Com a ajuda dos mapas digitais, o dispositivo indica-lhe a

²²aplicações com conhecimento do contexto

²³serviços *Web*

²⁴*Platform for Privacy Preference* - plataforma para as preferências de confidencialidade

forma de chegar a um destino previamente seleccionado. Para além disso, o turista poderá saber se existe um restaurante chinês por perto ou onde poderá ver um concerto de jazz, nessa mesma noite.



Figura 2.9: CRUMPET

Quando um utilizador se liga ao sistema, o dispositivo móvel centra-se no agente acessível na rede, esteja ele onde estiver. Esses agentes procuram reconhecer as palavras introduzidas, e seleccionam os serviços locais correspondentes. O sistema “aprende” os interesses e preferências dos utilizadores automaticamente. Por fim, o seu dispositivo precisa de adaptar os serviços à localização actual do utilizador. O sistema agente permite a parceiros como postos de Turismo, hotéis, organizadores de eventos e outros, que apresentem a informação de uma forma amigável e mantenham essa informação actualizada sem grande esforço. A plataforma pode ser utilizada em diferentes infra-estruturas como o *UMTS*, redes *wireless* ou redes locais (WLAN).

2.4.3 Cruso

O Cruso²⁵ é uma solução que implementa software e o seu próprio hardware, que tem como principais componentes de navegação um sistema GPS e uma bússola electrónica, que em conjunto permitem determinar a direcção para a qual o utilizador se encontra virado, determinando assim o contexto do utilizador. O turista pode tanto explorar uma cidade ou região por iniciativa própria como escolher um percurso pré-definido. Os pontos de interesse próximos do turista são assinalados no ecrã por um círculo (figura 2.10). O semi-círculo representa o campo de visão do turista. Quando o turista se vira para outra

²⁵<http://www.mycruso.de>

direcção a imagem do dispositivo ajusta-se dando sempre um mapa orientado relativamente à direcção do turista.

Para obter informação sobre um POI específico, basta seleccioná-lo no ecrã e será exibida a respectiva informação. O dispositivo fornece ainda informação em formato áudio com dois níveis de detalhe. Esta solução também permite a aquisição de bilhetes e pagamentos electrónicos através de ligações *wireless* seguras a curta distância.



Figura 2.10: Cruso²⁶

2.4.4 CityHelp

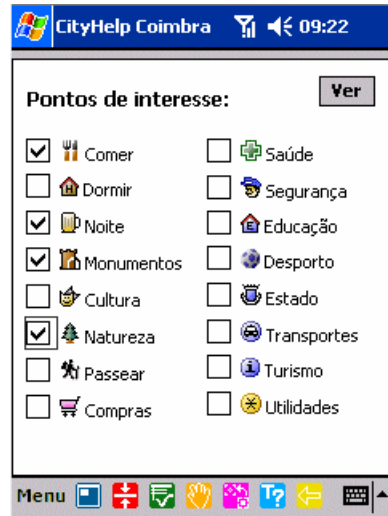
O CityHelp²⁷ é um roteiro digital e fornece aos seus utilizadores mapas de várias cidades portuguesas com informação detalhada dos POIs. Este sistema permite ao utilizador a pesquisa de ruas ou POIs. Com a função “estou aqui”, o utilizador pode conhecer todos os POIs mais próximos, e visualizá-los no mapa. O CityHelp fornece informação descritiva e fotográfica organizada em 16 categorias tais como comer, dormir, monumentos ou compras (figura 2.11). A posição pode ser facultada manualmente ou através de GPS. No caso da inserção manual, o utilizador pode na fase antes da viagem antever no mapa um trajecto que projecte realizar. Esta solução não necessita de estar ligada a qualquer rede móvel, ou de dispor de GPS, e pode ser instalada em PDA's ou telemóveis

2.4.5 Comparação dos sistemas

Na tabela 2.5 apresenta-se uma comparação geral dos guias discutidos em 2.4, sendo focados os seguintes pontos:

- Origem – O tipo de abordagem.

²⁷<http://www.cityhelp-pda.com>

Figura 2.11: CityHelp²⁸

- Arquitectura – Se a solução necessita de utilizar os recursos internos do dispositivo móvel, se necessita recorrer a serviços externos, ou ambos.
- Suporte – O tipo de dispositivo móvel utilizado.
- Objectivo Principal – O principal objectivo do guia.
- Aplicações – A aplicação da solução.

Guias Turísticos	Origem	Arquitectura	suporte	Objectivo Principal	Aplicação
COMPASS	sistema de recomendação dependente do contexto	interna externa	PDAs	sistema integrado baseado no contexto e sistemas de recomendação em aplicações móveis de Turismo	guia turístico baseado no contexto
CRUMPET	serviços móveis personalizados para Turismo	interna externa	PDAs	suporte a utilizadores móveis com serviços	guia turístico baseado na localização
Cruso	serviços baseados no contexto	interna	Hardware próprio	gerar informação turística baseada no contexto	guia turístico baseado na localização
CityHelp	serviços baseados na localização	interna	PDAs e telemóveis	suporte a visitante em cidades	roteiro turístico personalizado

Tabela 2.5: Comparação geral entre guias turísticos móveis [27]

Na tabela 2.6 é feita a comparação desses sistemas segundo os seguintes critérios:

- Obtenção da Posição – A forma como é obtida a posição do utilizador.
- Ciclo de vida – A fase do ciclo de vida do turista, no qual o guia serve o seu utilizador.
- Tipo de informação – A informação facultada pela aplicação

- Percurso – Se tem suporte a percurso pré-defenidos.
- Suporte a navegação – O tipo de suporte a navegação oferecido ao utilizador
- Contexto – O tipo de contexto suportado pela solução:
 - Localização
 - Tempo – se oferece suporte temporal
 - Dispositivo – suporte ao contexto orientação do dispositivo
 - Rede – suporte ao contexto da rede de telecomunicações
 - Utilizador – se oferece suporte ao perfil do utilizador

Guia	Obtenção Posição	Ciclo de vida			Tipo de informação	Percurso	Suporte de navegação	Contexto				
		Antes	Durante	Depois				Localização	Tempo	Dispositivo	Rede	Utilizador
COMPASS	GPS e redes móveis		x		Informação multimédia (texto, imagens) sobre os POIs tais como museus, monumentos, etc. Estes são marcados no mapa como símbolos. Os POIs podem facultar novos serviços		posição do utilizador no mapa	x	x			x
CRUMPET	GPS		x		Informação multimédia (texto, imagens e vídeos) sobre POIs tais como restaurantes, hotéis; na forma de serviços de recomendações ou dicas pró-activas. POIs da categoria seleccionada são exibidos no mapa como símbolos		o caminho mais curto entre a posição do utilizador e o POI	x		x	x	x
Cruso	GPS e bússola electrónica		x		informação multimédia (texto, imagens e áudio) sobre POIs tais como monumentos, restaurantes. Serviços como bilhetes e pagamentos electrónicos	pré-definido	posição do utilizador	x		x		
CityHelp	GPS ou manual	x	x		informação multimédia (texto e imagens)		caminho mais curto	x				

Tabela 2.6: Comparação geral entre guias turísticos móveis [27]

As soluções apresentadas centram-se mais na fase do decorrer da viagem, apresentam várias funcionalidades a nível de localização, temporal, rede, dispositivo e utilizador. Todos os guias apresentam suporte ao contexto de localização, alguns oferecendo serviços como compra de bilhetes ou reservas de lugares em restaurantes. O CRUMPET e o COMPASS têm suporte a perfis e é nesse aspecto que o COMPASS apresenta suporte ao contexto temporal, e nas recomendações que fará ao utilizador, dando menor importância à categoria que o utilizador tenha visitado recentemente. A nível de informação, estas soluções ostentam diversos conteúdos de multimédia como fotos, áudio e vídeo. O CityHelp apresenta uma total autonomia em relação a ligações às redes de comunicações. Este tipo de soluções são ainda recentes, e previsivelmente nos próximos anos tornar-se-ão cada vez mais complexas e completas.

2.5 Sumário

Neste capítulo, começa-se por falar da importância do Turismo na economia portuguesa, o qual se prevê que venha a representar 6,4% do PIB e empregue directa e indirectamente cerca de 19% da população activa, tornando este sector económico muito atractivo. Introduzem-se alguns conceitos e definições sobre Turismo e discute-se a importância da informação no Turismo. Contemplando essa necessidade de informação, colocam-se as questões que se levantam no desenho de aplicações ubíquas com realce nas dimensões *o quê*, *quando*, *onde*. Introduz-se e define-se o termo serviço baseado na localização, e mostra-se que estes serviços têm por base os GISs, a *Internet* e as novas tecnologias de informação e de telecomunicações. Introduzem-se os serviços *Pull* e *Push*. Depois fala-se nos requisitos e nos componentes necessário numa arquitectura LBSs e termina-se este assunto com alguns exemplos de implementações.

Em seguida, introduzem-se as BDs temporais com alguns conceitos relevantes. Prossegue-se com o XML temporal no qual se discute a implementação de tempo de validade em documentos *Web* com a ajuda do XML. Para a consulta de documentos XML, utiliza-se a linguagem de interrogação XQuery e são dados alguns exemplos de consultas temporais.

Por fim, termina-se o capítulo com a discussão de algumas soluções já implementadas de guias turísticos móveis.

Capítulo 3

Modelação Conceptual e Implementação do Sistema

No capítulo anterior, discutiu-se a importância da informação no Turismo e a necessidade facultar aos turistas novos serviços que facilitem o acesso a essa informação onde e quando lhes é necessária. Os sistemas de informação (SI) turística ou guias turísticos para dispositivos móveis são recentes e certamente terão um grande desenvolvimento nos próximos anos.

Neste capítulo faz-se uma descrição de um modelo para o desenvolvimento de um SI turística para suporte à implementação de guias turísticos acessíveis via PDAs. O foco deste trabalho incide numa proposta de modelo para a representação de informação sobre a disponibilidade de recursos turísticos e a aplicação dessa informação na elaboração de percursos turísticos. O sistema deve assentar numa arquitectura de cliente-servidor. Os clientes efectuarão os pedidos ao servidor. Esses pedidos serão processados pelo servidor e este enviará as respostas novamente para os clientes.

3.1 Descrição do sistema

Segundo o trabalho de Khai Truong, apresentado na secção 2.2.3, existem cinco dimensões principais, que devem ser consideradas no desenho de aplicações ubíquas.

Na dimensão de **“quem”**, deve-se identificar os intervenientes e quais as tarefas que desempenham no sistema. No caso deste SI turístico, existe um administrador do sistema, que tem como funções supervisionar o sistema e garantir o seu bom funcionamento. O agente turístico é responsável pela introdução e actualização da informação relativa aos POIs e percursos turísticos pré-definidos. O turista poderá fazer a introdução e actualização do seu perfil e terá acesso à informação sobre os POIs e percursos turísticos disponibilizados pelos agentes.

A dimensão **“o quê”** descreve a informação a armazenar na BD e será composta pelos

seguintes tipos de informação:

Temática: Esta informação deve estar organizada por temas por forma a facilitar as operações de pesquisa. Deve identificar os POIs com a sua designação, endereço, contactos, ter uma descrição geral resumida e disponibilizar informação detalhada por forma a satisfazer as necessidades de diversos tipos de turistas durante as suas viagens. Detalhes sobre acessibilidades para pessoas com deficiência, informações sobre os preços (exactos ou indicativos, classificando o nível de preços de uma a cinco estrelas), bens e serviços, são bons exemplos de informações úteis a considerar neste tipo de sistemas.

Multimédia: Esta informação deve permitir visualizar com fotos com os locais de visita, ou outras imagens com interesse para os turistas. Os vídeos permitem explicar e focar aspectos mais importantes do POIs. Deve ainda fornecer endereços da *Internet* relacionados com o POI de forma a obter informação adicional.

A dimensão “**onde**” deverá considerar a informação necessária para localizar de forma eficaz os POIs com recurso a coordenadas geográficas, permitindo a sua utilização tanto para a criação de percurso como a sua visualização em mapas geográficos.

A dimensão “**quando**” deve contemplar a informação temporal sobre a disponibilidade do POI ou serviço, assim como facultar a validação dos dados na criação de percursos pré-definidos.

A informação temporal disponibilizada deverá representar horários de funcionamento ou a disponibilidade de serviços. Tipicamente, os horários são representados no decorrer de uma semana, ou seja, são apresentados os períodos diários em que o POI está aberto. Esses períodos são formados por um dia ou um grupo de dias com períodos de horas do(s) dia(s) associados, como apresentado no exemplo 3.1.

Exemplo 3.1 *Horário do Departamento de Electrónica, Telecomunicações e Informática.*

De segunda a sexta: das 8h30 às 20h00 e das 20h30 as 23h00.

Aos sábados: das 10h00 às 13h00.

Existem ainda os feriados, festividades ou férias que ocorrem normalmente em determinadas datas, podendo abranger um ou vários dias.

A dimensão de “**como**”, determinar como usar os serviços. O administrador e os agentes turísticos devem poder utilizar o sistema em qualquer local com acesso à *Internet*. Tipicamente deverão poder utilizar o sistema a partir de qualquer computador com ligação à *Internet* e disponibilizar os conteúdos no servidor. No caso dos turistas, deve-se proceder de outra forma, pois estarão munidos de PDAs e poderão estar em

locais sem ligação à *Internet*, devendo no entanto ser capazes de utilizar o sistema. Para tal, devem descarregar a informação da BD para os seus dispositivos, tornando-se assim autónomos.

Os percursos turísticos pré-definidos permitem mostrar aos turistas os pontos mais importantes de uma cidade ou de uma região, eventualmente organizados por temas. Esses percursos devem ser elaborados por forma a abranger e cativar a maior parte dos turistas.

Para a utilização deste sistema, o turista tem de descarregar os ficheiros com a informação sobre os POIs, serviços e percursos pré-definidos. Depois de descarregados, o turista poderá consultar toda a informação, incluindo os horários de funcionamento dos POIs ou serviços. Cabe aos agentes e aos parceiros turísticos a tarefa de introdução e sua actualização dos conteúdos. Os percursos pré-definidos são mantidos pelos agentes turísticos.

Os percursos são elaborados mediante um tema que pode ser escolhido segundo os POIs da zona que se quer dar a conhecer. No seu planeamento, os agentes turísticos devem localizar os diferentes recursos, os percursos possíveis e analisar as distâncias, os tempos e os custos. Mediante esses factores deve determinar o percurso final e então armazenar os percursos na BD, tornando-os acessíveis aos turistas.

3.2 Arquitectura do sistema

A arquitectura do sistema é do tipo cliente/servidor. Na lado do cliente, encontram-se os agentes turísticos que acedem ao sistema via *browser* e os turistas via PDAs. Os agentes turísticos são responsáveis pela introdução e actualização da informação na BD. No *browser*, os agentes inserem a informação em formulários que enviam ao servidor *Web* para actualização da BD (figura 3.1). Quando o agente cria um percurso pré-definido, a informação também precisa de ser validada (temporalmente) na BD. Assim, a cada iteração da criação do percurso a informação é enviada ao servidor que a valida e envia o resultado para o *browser* do agente turístico. Ao terminar o percurso, este é armazenado na BD. O turista por sua vez, quando se liga ao sistema, o seu perfil é enviado à BD e, no caso de ter sido alterado, é actualizado. Depois o turista pode descarregar toda a informação (temática, multimédia, temporal, geográfica e percursos pré-definidos) da BD. A partir desse momento o turista age perante o sistema de forma autónoma (figura 3.2).

O tema desta dissertação está associado a um sistema já em desenvolvimento que contempla a manutenção da informação sobre o perfil dos utilizadores e as componentes temática, multimédia e geográfica sobre os POIs. Desta forma, o trabalho realizado incide no desenvolvimento dos aspectos temporais relacionados com a disponibilidade dos recursos, serviços ou eventos com interesse turístico e a elaboração de percursos turísticos pré-definidos, tal como indicado a sombreado na figura 3.1.

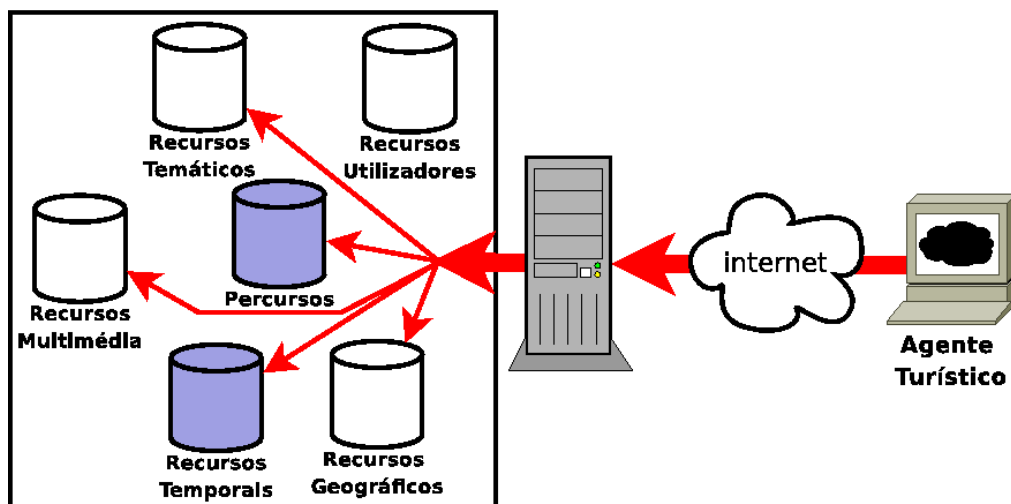


Figura 3.1: Arquitectura do sistema - o agente turístico

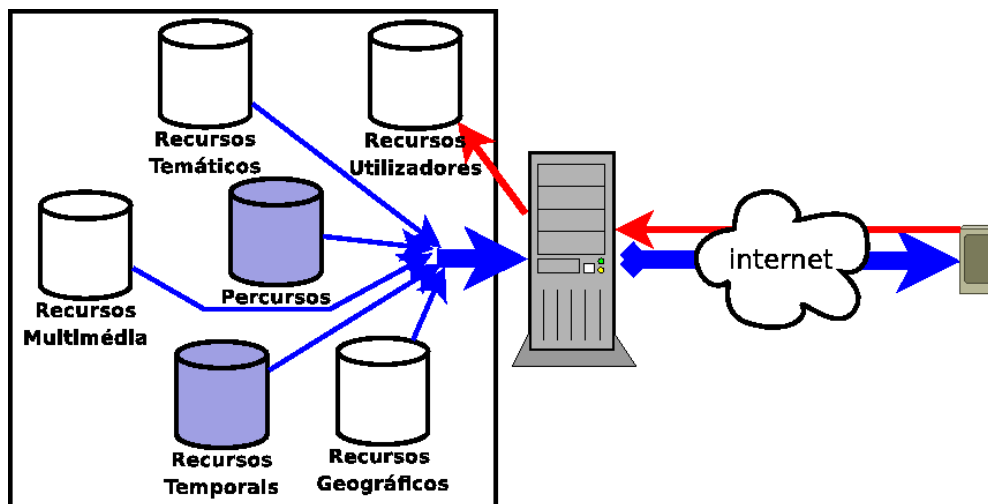


Figura 3.2: Arquitectura do sistema - o turista

3.3 Modelo conceptual do sistema

3.3.1 Requisitos

Funcionais

- Desenhar e implementar uma extensão ao modelo de dados existente contemplando os aspectos temporais relevantes num SI turístico nomeadamente:
 - definir os horários de funcionamento dos serviços nos formatos habitualmente disponibilizados aos consumidores. Por exemplo, o modelo de dados temporal deverá ser suficientemente flexível para indicar que um restaurante está aberto de 3^a-feira a domingo; que durante a semana está aberto das 12 às 23 horas e ao fim-de-semana está aberto das 12 às 24 horas; e que encerra para férias no mês de Outubro.
 - definir as datas e horários de eventos pontuais, por exemplo, a realização de uma prova desportiva importante, eventos sazonais ou cíclicos, por exemplo, as festividades dos Santos Populares.
 - conceber e analisar diferentes cenários de utilização de um SI turístico para a elaboração de percursos turísticos.

Não-funcionais

Autonomia - O turista deve ter autonomia em relação ao SI turística. Os locais onde se poderá encontrar nem sempre facultarão uma ligação à *Internet* e quando isso acontece o turista deve manter capacidade de utilizar o sistema.

Portabilidade - O sistema deverá ser portátil, isto é, os utilizadores poderão utilizar o sistema através de qualquer sistema operativo e através de qualquer *browser*.

Interoperabilidade - Este sistema será usado por utilizadores com finalidades diferentes. Os agentes turísticos deverão fornecer a informação turística, temporal e geográfica, que deve ficar disponível para o turista a poder utilizar, descarregando-a para os seus dispositivos móveis. Além disso, os agentes têm de aceder à informação da BD por forma a criarem novos percursos pré-definidos

3.3.2 Modelo de domínio do sistema

O modelo de dados herdado já contempla a informação temática, multimédia e geográfica. Na figura 3.3 apresenta-se o modelo de domínio do sistema e a sombreado encontra-se a parte desenvolvida e acrescentada neste projecto. De forma sucinta, passa-se a explicar o modelo.

- Os utilizadores podem ser agentes turísticos ou turistas. Para ambos, existe um identificador e uma password.
 - O turista terá um perfil que é formado com os seus dados pessoais e pelas suas preferências, tem uma localização e pode visualizar a sua posição no mapa. O turista poderá visualizar os vários percursos pré-definidos, assim como os POIs e serviços.
 - O agente turístico, como por exemplo um operador turístico, poderá ter um ou mais POIs ou serviços associados. O operador pode visualizar e actualizar a informação sobre os respectivos POIs ou serviços ou criar vários percursos turísticos.
- Os POIs estão classificados de acordo com as 5 categorias indicadas na figura 3.4.
- Os serviços estão classificados de acordo com as 6 categorias indicadas na figura 3.5.
- Os serviços e os POIs têm uma única localização, e esta encontra-se numa zona coberta pelos mapas guardados no sistema.
- Os serviços ou POIs poderão ter vários recursos multimédia, como fotos, vídeos ou páginas Web relacionadas.
- Os POIs ou serviços têm um horário.
- Um POI pode estar incluído num ou mais percursos pré-definidos, o mesmo acontecendo para os serviços. Um percurso pré-definido pode incluir vários serviços ou POIs associados.
- um percurso pré-definido pode ter zero ou mais recursos multimédia.

Neste trabalho introduz-se uma forma de tratar das necessidades temporais. A nível do modelo de domínio do sistema, aparece a classe horário que permitirá representar a disponibilidade de um POI ou serviço.

3.4 Desenho e implementação

3.4.1 Repositório de dados

Como foi referido na secção 2.3.2, o XML tem grandes vantagens na troca de informação através da *Internet*. Neste projecto essa característica é deveras importante porque facilita a interoperabilidade entre os agentes que introduzem a informação no servidor e os turistas que descarregam a informação do servidor para os seus PDAs. O seu processamento também é mais leve, e no caso dos PDAs revela-se num aspecto relevante, apesar do crescente poder de processamento que estes têm mostrado.

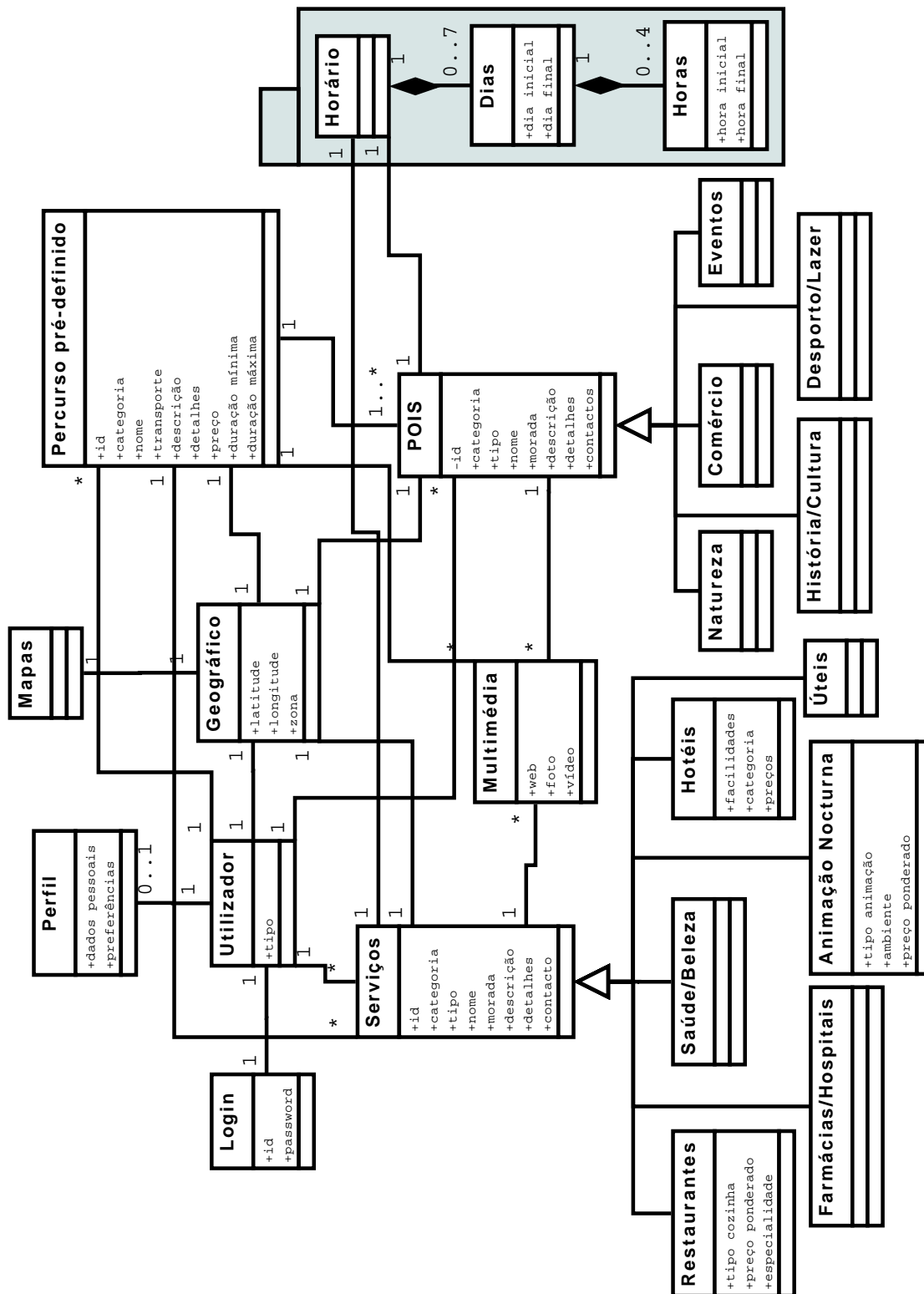


Figura 3.3: modelo domínio

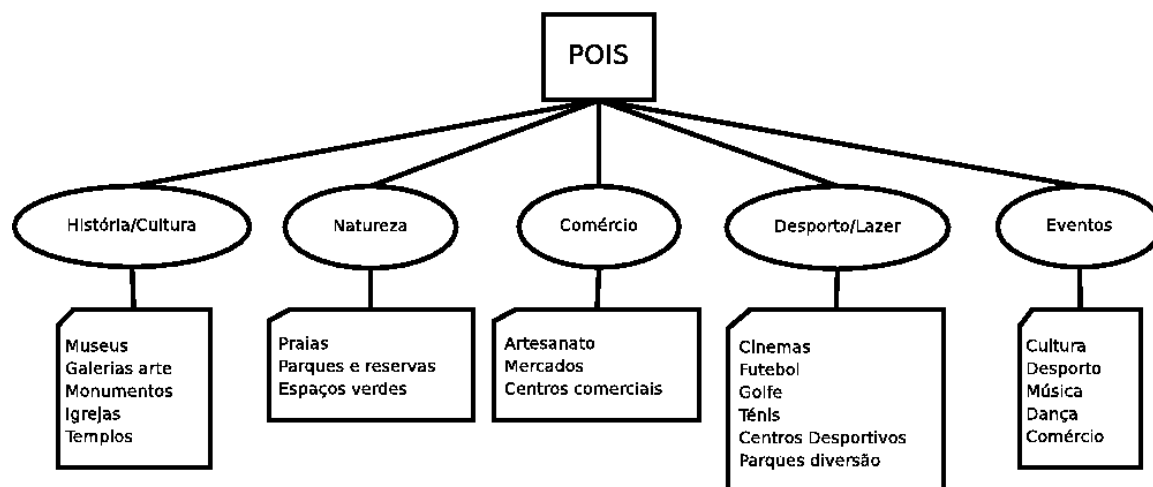


Figura 3.4: Categorias e tipo de POIs

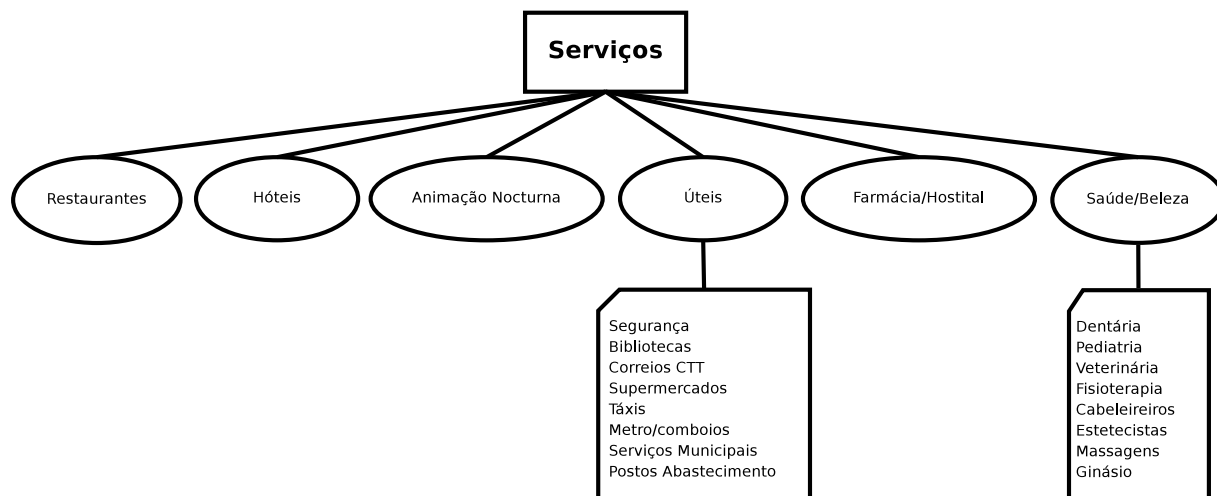


Figura 3.5: Categoria e tipo de serviços

Assim sendo, toda a informação é armazenada em ficheiros XML. Os perfis dos utilizadores ficam armazenados no ficheiro “registos.xml” e os percursos pré-definidos ficarão guardados no ficheiro “rotas.xml”. Os recursos multimédia encontram-se alojados em duas pastas, fotos e vídeos. A informação geográfica esta dividida em duas partes, os mapas e as coordenadas geográficas referentes às localizações dos POIs e serviços. Os mapas, da mesma forma que os recursos multimédia são armazenados numa pasta, e as coordenadas são armazenadas nos registos dos respectivos POIs ou serviços. A informação temporal também é guardada nos registos de cada POI ou serviço. Na tabela 3.1 encontram-se os elementos que se pode encontrar nos registos dos POIs ou serviços.

Elemento		tipo de dados	Descrição
id	P/S	inteiros	identificador do POI ou serviço, este é utilizado internamente.
category_type	P/S	enumerado (figura 3.4)	Categoria
type	P/S	enumerado (figura3.4)	Tipo subdivisão da categoria
name	P/S	texto	Designação
address	P/S	texto	Morada
latitude	P/S	reais em $[-90, 90]$	Referente a informação geográfica
longitude	P/S	reais em $[-180, 180]$	Referente a informação geográfica
zone	P/S	texto	Zona onde se encontra o POI ou serviço
description	P/S	texto	descrição geral do POI ou serviço
details	P/S	texto	Informação mais detalhada
contact	P/S	inteiros	Os contactos telefónicos
time_table	P/S	texto	O horário de funcionamento.
type_cuisine	S	texto	No caso de restaurantes, este atributo classifica-o no tipo de cozinha
weighted_price	S	texto	Nos serviços, preço médio ou ponderado por refeição
speciality	S	texto	No caso dos restaurantes, poder realçar um prato
price	P	texto	preço quando aplicável, por exemplo preço de entrada
web	P/S	texto	endereço(s) <i>Internet</i>
photo	P/S	texto	nome dos ficheiros imagem relacionados
video	P/S	texto	nome dos vídeos relacionados

Tabela 3.1: Elementos dos POIs e serviços

Os elementos categoria e tipo servem para classificar os POIs e serviços e limitar as pesquisas em termos da informação temática. A zona serve para limitar as pesquisas geograficamente numa determinada área. Alguns dados são sujeitos a multiplicidade, ou

seja, podem ter mais que um valor válido, por exemplo, os contactos, as fotos ou os vídeos. Neste caso, os elementos albergam todos os valores separados pelo símbolo “|”.

Elemento	Descrição
id	Identificador do percurso. Os valores que toma são inteiros
type	Tema do percurso pré-definido pode tomar valores das categorias dos POIs.
name	Nome do percurso
transport	Meio de transporte a utilizar para realizar o percurso
description	Descrição geral do percurso
details	Detalhes do percurso
duration_min	Estimativa da duração mínima prevista para a realização do percurso.
duration_max	Estimativa da duração máxima prevista para a realização do percurso
cost	Estimativa do custo da realização do percurso
POIS	Identificação dos POIS sugeridos no percurso
photo	Nome dos ficheiros imagem relacionados com o percurso
video	Nome dos vídeos relacionados com o percurso
map	nome do mapa do percurso

Tabela 3.2: Rotas.xml

Após a elaboração dos percursos pré-definidos, estes serão armazenados no ficheiro “Rotas.xml”. O elemento *transport* serve de restrição. Os elementos “duration_min” e “duration_max” são expressos em horas. No elemento POIS, cada POI do percurso aparece numa ordem e separado pelo simbolo “|”. Essa ordem indica a sequência dos POIs a visitar.

Nos registos dos utilizadores, as preferências podem tomar os valores “sim” no caso do tema ser da sua preferência e “não” em caso contrário.

O trabalho de Fábio Grandi na implementação temporal de uma infra-estrutura para a gestão temporal de documentos Web [24] que foi abordado em 2.3.2 foi a base na elaboração deste trabalho. Com a estrutura utilizada por Grandi e recordando os conceitos de tempo introduzidos em 2.3.1, um instante de tempo é representado na forma apresentada no exemplo 3.2. O exemplo 3.3 define um intervalo de tempo e no exemplo 3.4 descreve-se um elemento de tempo.

Exemplo 3.2 Representação de um instante de tempo

```
<valid>
  <validity from="1980-01-01" to="1980-01-01"/>
  Este texto é valido no dia 1 de Janeiro de 1980.
  Assim obtem-se um instante de tempo.
</valid>
```

O instante de tempo é representado com a mesma data nos atributos from e to

Exemplo 3.3 Representação de um intervalo de tempo

Elemento	Tipo de dados	Descrição
User	texto	“login” do utilizador
Password	texto	Palavra-chave
Name	texto	Nome completo do utilizador
Age	inteiro	Idade do utilizador
Genre	caracteres	Sexo
Country	texto	País de residência
Mail	texto	Endereço de correio electrónico
Pref_History_Culture	texto	Se o utilizador tem preferência em temas relativos à história e cultura
Pref_Nature	texto	Se o utilizador tem preferência em temas relativos à natureza
Pref_Trade	texto	Se o utilizador tem preferência em temas relativos ao comércio
Pref_Entertainment	texto	Se o utilizador tem preferência em temas de entretenimento
Pref_Sport	texto	Se o utilizador tem preferência no tema do desporto

Tabela 3.3: Registos.xml

```
<valid>
  <validity from="1980-01-01" to="1980-02-15"/>
  O presente texto é válido no intervalo de dia 1 de Janeiro de 1980 a 15
  de Fevereiro de 1980.
</valid>
```

Exemplo 3.4 Representação de um elemento de tempo

```
<valid>
  <validity from="1980-01-01" to="1980-01-01"/>
  <validity from="1980-02-05" to="1980-02-13">
    Este texto é válido no dia 1 de Janeiro e no intervalo de dia 5 de Fe-
    vereiro a dia 13 do mesmo mês.
  </validity>
</valid>
```

A nova estrutura também deverá implementar estas representações. Para o tratamento da informação temporal, foi necessário criar uma estrutura em XML e aglutina-la à existente, nos ficheiros “POIS.xml” e “serviços.xml”. Essa estrutura está representada na figura 3.6, podendo-se verificar que é representada por um elemento raiz que será `time_table` e conterá até 7 elementos `day`, estes por sua vez poderão conter até 4 elementos `hour`. O exemplo 3.5 ilustra o horário da biblioteca.

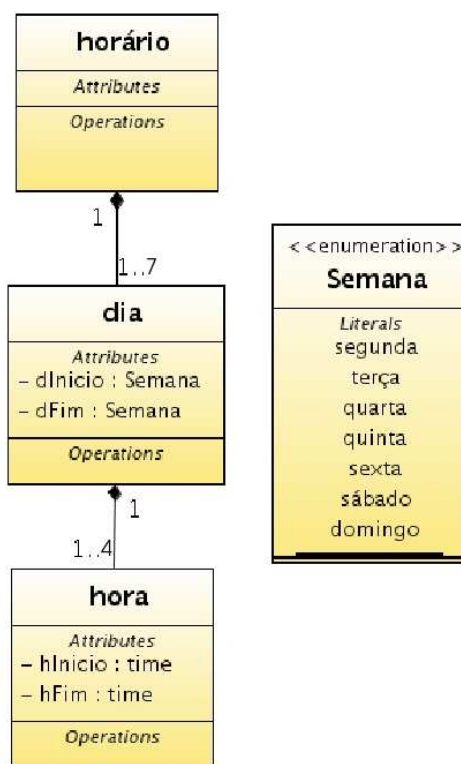


Figura 3.6: Diagrama de classes Horário

Exemplo 3.5 *Horário de um Biblioteca*

```

<time_table>
  <day dStart="monday" dEnd="friday">
    <hour hStart="09:00" hEnd="22:00"/>
  </day>
  <day dStart="saturday" dEnd="saturday">
    <hour hStart="10:00" hEnd="13:00"/>
  </day>
</time_table>

```

A partir do exemplo 3.5 verifica-se que para o caso do dia de sábado, que o atributo `dEnd` é o mesmo que `dStart`, isto significa que esse elemento define o horário apenas para esse dia. No caso das horas, os dois atributos são obrigatórios. Esta representação permite definir um horário diferente para cada dia da semana.

Com esta nova estrutura, é possível representar os diversos tipos de horários especificados nos requisitos do sistema. Permite representar até um horário para cada dia da semana, o que muito raramente deverá acontecer e em cada dia poderá ser representado até 4 períodos de abertura.

Para a implementação desta estrutura foi necessário criar um *Schema* XML que se passa a explicar. Para começar foi criado um enumerado com os nomes dos dias da semana ao qual se dá o nome de `week` e é definido no código 3.1.

Código 3.1 Enumerado da semana

```
<xs:simpleType name="week">
  <xs:restriction base="xs:string">
    <xs:enumeration value="monday"/>
    <xs:enumeration value="tuesday"/>
    <xs:enumeration value="wednesday"/>
    <xs:enumeration value="thursday"/>
    <xs:enumeration value="friday"/>
    <xs:enumeration value="saturday"/>
    <xs:enumeration value="sunday"/>
  </xs:restriction>
</xs:simpleType>
```

Depois de definido o enumerado, é possível definir ao resto do *schema* apresentado em código 3.2, e definem-se o elemento `time_table` como elemento raiz, que poderá ter no máximo 7 elementos `day` e os seus atributos serão do tipo enumerado criados mais atrás. Os elementos `dia` poderão conter no máximo 4 elementos `hour` que por sua vez terão os seus atributos de um tipo pré-definidos pelo W3C, `xs:time`. Como se pode ver no código todos os atributos são obrigatórios

Com este *Schema* XML será garantido a estrutura para os elementos dos horários de funcionamento dos POIs ou serviços.

Para a introdução destes horários, foi criada uma interface flexível (figura 3.7). Na interface, aparecem menus **dropdown** com dia inicial e dia final. Os valores que estes poderão tomar são o nome dos dias da semana e portanto formarão um período de dias. Os atributos “hora inicial” e “hora final” definem o período de horas de funcionamento para os respectivos dias. Para adicionar mais períodos, o utilizador deverá clicar no botão com o sinal “+” aparecendo dois novos campos para o efeito. Se o horário não for uniforme ao longo da semana, poderá adicionar novos dias clicando no botão “adicionar dia” e surgirá uma estrutura idêntica à inicial.

Contudo esta estrutura, tal como a de Grandi, define apenas elementos de tempo em que se verifica a efectiva abertura e não contempla, casos de excepção, como por exemplo a biblioteca está aberta todos os dias menos ao Domingo. Este caso deverá ser visto da forma, a biblioteca está aberta de Segunda a Sábado. Um caso não contemplado é o de períodos de excepção como “a biblioteca está fechada no período de 22 de Dezembro a 2 de Janeiro”.

Código 3.2 Schema XML para os horários

```
<xs:element name="time_table">
  <xs:complexType>
    <xs:sequence maxOccurs="7">
      <xs:element name="day">
        <xs:complexType>
          <xs:sequence maxOccurs="4">
            <xs:element name="hour">
              <xs:complexType>
                <xs:attribute name="hStar" type="xs:time" use="required"/>
                <xs:attribute name="hEnd" type="xs:time" use="required"/>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="wStart" type="week" use="required"/>
          <xs:attribute name="wEnd" type="week" use="required"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Figura 3.7: interface de introdução de horários

3.4.2 Métodos de processamento dos dados

Uma vez apresentadas as estruturas de dados para a representação de dados temporais, importa agora definir os métodos específicos que foram desenvolvidos para o processamento deste tipo de dados, que serão o suporte à implementação de consultas usando critérios temporais. Esses métodos foram desenvolvidos em linguagem XQuery. Apresentam-se de seguida os processos mais relevantes realizados no servidor.

Como os atributos dos elementos `day` são do tipo enumerado `week` e na interface desenvolvida se opera com datas, era necessário transformar essas datas de modo a obter os dias da semana:

```

day-of-week($date as xs:anyAtomicType?) as xs:integer?
dayWeek($i as xs:integer?) as xs:string?
indSem($i as xs:string) as xs:integer?
dayOfDate($d as xs:date?) as xs:string?

```

A primeira função, dada uma data devolve o índice do dia da semana, 1 para segunda, 2 para terça e assim sucessivamente. Esta função é retirada de um repositório *online* de funções XQuery [26]. A segunda função, devolve o nome do dia da semana dado o índice devolvido pela primeira função. A penúltima permite fazer a operação inversa à função *dayWeek*, por fim a função “dayOfDate” aceita uma data e devolve o dia da semana. No entanto também era necessário poder determinar se, por exemplo, “wednesday” estava contido num intervalo [“monday”, “friday”]. A próxima função foi desenvolvida com essa finalidade e aceita um elemento `day` e uma `string` contendo o dia a verificar, o seu protótipo é

```

containDay($i as item(), $s as xs:string) as xs:boolean

```

Essa função devolve um tipo booleano. As funções até agora manipulam datas e dias da semana e são úteis para manobrar os elementos `day`. Para tratar os elementos `hour`, foram criadas funções similares tais como:

`containHour($i as item(), $h as xs:time) as xs:boolean`

A função `containDay` aceita como argumentos um elemento `hour` e um instante de tempo `$h`, verifica se esse instante está contido no intervalo formado pelo elemento, e devolve um booleano como resultado.

`doubToDur($d as xs:double?) as xdt:dayTimeDuration`
`strToDur($s as xs:string?) as xdt:dayTimeDuration`
`doubToTime($d as xs:double) as xs:string`

A função `doubToDur` recebe como argumento um valor do tipo `double` e converte-o num do tipo `dayTimeDuration` que permitirá representar períodos de tempo. A função `strToDur` aceita uma `string` na forma “HH:MM” e transforma-a no tipo `dayTimeDuration`. Esta função serve para converter as *strings* lidas nos elementos `hour` dos ficheiros XML da BD para um período de maneira a operar com o tempo. Por fim a função `doubToTime` transforma um `double` numa *string* da forma “HH:MM” podendo dessa maneira armazenar nos elementos `hour`, valores nessa forma.

A próxima função verifica, dados o id dum registo de POI ou serviço, um dia da semana e uma hora, se está aberto e devolve ‘*true*’ em caso afirmativo e ‘*false*’ em caso contrário.

`dayOpen($id as xs:integer, $dia as xs:string, $hora as xs:time) as xs:boolean`

3.5 Sumário

Neste capítulo, começou-se por descrever o sistema contextualizando-o com o trabalho de Khai Truong, que define cinco categorias de questões a colocar no desenvolvimento de aplicações ubíquas: o quê? Quem? Quando? Onde? Como? Assim identificaram-se dois utilizadores principais do sistema, o agentes turístico e o turista. Existe ainda o administrador que assume meramente tarefas de gestão do sistema. Quanto às tarefas dos agentes turísticos, são de inserção e actualização dos conteúdos temáticos, geográfico, multimédia e temporal dos serviços e POIs e, são ainda responsáveis pela elaboração de percursos temáticos pré-definidos. Toda a informação atrás referida tem como propósito ser acessível ao turistas. Os turistas têm a tarefa da utilização dessa informação onde quer que se encontrem. Isto implica que mesmo não tendo ligação a uma rede móvel ou *wireless*, deverão continuar a usufruir dos serviços. Para que tal aconteça, os turistas deverão descarregar a informação para os seus dispositivos móveis.

No desenvolvimento, discutiu-se a escolha do XML para a implementação e as vantagens que auferir na transferência de dados pela Internet permitindo uma grande interoperabilidade entre os utilizadores. Em seguida, apresentou-se uma infra-estrutura para a gestão temporal de documentos Web baseada no trabalho de Grandi, para suportar a implementação dos novos os horários de funcionamento dos serviços.

Capítulo 4

Protótipo para a Elaboração de Percursos Turísticos

O capítulo anterior apresenta as etapas que levaram à criação de um SI turística orientado para dispositivos móveis. Esta informação, para além de poder ser disponibilizada directamente para consulta aos turistas, também serve de base à elaboração de percursos turísticos temáticos. Os serviços disponibilizados pelos SBL apresentados no capítulo 2 normalmente têm como ponto de partida a localização do turista num dado momento, estando orientados para responder a questões do tipo “indicar quais os restaurantes próximos da localização do turista”. No SI turística que serve de enquadramento a esta dissertação também se pretende disponibilizar um conjunto de percursos temáticos pré-definidos.

Para a elaboração destes percursos, a localização do turista num dado instante perde relevância cedendo lugar à noção de trajecto entre um ponto de partida e um ponto de chegada. Desta forma, existe um novo conjunto de variáveis como o tempo de deslocamento entre POIs, a ordenação das actividades ao longo de um período de tempo ou distância relativa a um trajecto em vez de posição fixa, que deve ser considerado. A proposta apresentada neste trabalho, baseia-se na elaboração de percursos turísticos pré-definidos de forma iterativa, em que o agente turístico localiza os diferentes recursos, identifica os percursos possíveis, analisa as distâncias, os tempos e os custos.

4.1 Funcionalidade esperada

A base de dados apresentada no capítulo 3 permite definir um conjunto diversificado de opções de pesquisa, tendo em vista satisfazer as necessidades de informação dos turistas e dos operadores turísticos em diferentes contextos de utilização.

1. Consultas baseadas em critérios geográficos ou temáticos:

- (a) “Onde fica o restaurante a Gruta?”

Neste caso, o turista deseja conhecer a localização de um POI, dado com atributo descritivo como uma categoria, tipo ou um nome, independentemente da

posição.

- (b) “Onde fica o restaurante mais próximo?”

O turista pretende conhecer o POI mais próximo da sua localização.

2. Consultas envolvendo critérios temporais e a noção de percurso:

- (a) “Estou nesta localização, vou a pé, e quero ir ao museu Sta Joana, depois quero passar pela farmácia Central e se tiver tempo ainda ir ao banco.”

Neste cenário, dados a localização, o meio de transporte a utilizar, os POIs a visitar assim como o tempo a dispensar para cada um deles, pretende-se criar de forma iterativa o percurso do utilizador, validando temporalmente cada iteração.

- (b) “Estou nesta localização, quero ver museus e estou a pé.”

Dada a localização do utilizador, o meio de transporte e tipo de POI, o utilizador deseja conhecer os POIs existentes ao alcance do meio de transporte utilizado.

- (c) “No dia 15/10, estarei no local A e irei para o local B, e quero ver qualquer assunto relacionado com história, natureza e comércio; quais os locais mais próximos relacionados?”

Dados um trajecto, uma data e um meio de transporte, o utilizador pretende conhecer POIs ao seu alcance durante o percurso.

- (d) “Estarei no local A do dia 13/11 ao dia 2/12, quero visitar museus ao sábado e pretendo deslocar-me a pé; que opções tenho?”

Dado um período de tempo, uma localização, uma restrição temporal e o meio de transporte, o utilizador pretende obter os POIs ao seu alcance que estão abertos.

4.2 Ferramentas utilizadas

Neste projecto recorreu-se a ferramentas disponibilizadas para o sistema operativo *linux* e todas são de domínio público. Para a base de dados, utilizou-se a *eXist-db*, uma base de dados nativa em XML que para além da possibilidade armazenar documentos XML, oferece três maneiras diferentes de processamento de *XQuery*, uma das quais permite guardar uma aplicação *Web* completa. Para o desenvolvimento das páginas *Web*, utilizou-se a linguagem HTML e Javascript. Esta última permite validar os dados introduzidos numa página *Web*. O browser que serviu de suporte para o desenvolvimento das páginas *Web* foi o *Firefox* que inclui algumas extensões úteis no desenvolvimento *Web*. Por fim, utilizou-se um editor de texto muito versátil com características interessantes, o *Kate*.

eXist-db

O **eXist-db** é um sistema de gestão de dados criado inteiramente em tecnologia XML. É um sistema nativo de código aberto que suporta várias tecnologias *Web* standard [31]:

- XQuery 1.0 / XPath 2.0
- interfaces HTTP: REST, WebDAV, SOAP, XMLRPC, Atom Publishing Protocol

eXist-db é largamente compatível com o standard XQuery (obteve a pontuação de 99,4% no actual XQTS¹). O motor de Xquery é extensível e possui uma grande colecção de módulo de funções XQuery. **eXist-db** fornece um ambiente de desenvolvimento de aplicações *Web* baseadas em XQuery e standard relacionados. Todas as aplicações *Web* podem ser escritas em XQuery, usando XSLT, XHTML, CSS e Javascript (para funcionalidades Ajax). XQuery server pages podem ser executadas a partir do sistema de ficheiros [31].

Para processar as xquery, o eXist-db utiliza o Saxon, um processador de XSLT e XQUERY de código fonte aberto. A sua última versão (9.1.0.2) é uma implementação completa e em conformidade com as recomendações de 23 de Janeiro de 2007 do W3C.

Esta solução não necessita de nenhum *software* adicional para correr e fornece um servidor de *WebServices* em código aberto denominado *Jetty* que pode correr como uma solução autónoma.

O eXist-db fornece 3 maneiras diferentes de chamar as XQuery através da Internet. Neste projecto optou-se por fazer pedidos através do servidor REST. Para tal, é utilizado o *servlet* REST que permite executar *stored XQueries* no servidor. Quando recebe um pedido HTTP *Request* com os métodos GET ou POST, o servidor REST tenta ler e executar esse pedido com o contexto do HTTP *Request*. Este é um conceito poderoso que permite guardar aplicações *Web* completas na BD [31].

Javascript

O *javascript* é uma linguagem *script* usada a maior parte das vezes no lado do cliente para o desenvolvimento de aplicações web. O *javascript* pode ser incluído em páginas HTML e interagir com o *Document Object Model* (DOM) da página. De uma forma muito simplista, o DOM considera um documento HTML como uma estrutura em árvore de elementos, atributos e texto. O DOM define os objectos e propriedades de todos os elementos HTML assim como métodos para aceder, alterar, criar e apagar esses objecto. A interacção do DOM com o *javascript* possibilita alterar a estrutura de páginas HTML dinamicamente. O javascript também permite a validação dos dados no lado do cliente permitindo dessa forma aliviar tarefas ao servidor.

Mozilla Firefox 3.0

O Mozilla Firefox é um browser criado pela Mozilla, que também desenvolveu o Netscape. O projecto Firefox teve inicio em Novembro de 2004, tendo uma ascensão muito rápida. O facto de ser um projecto com código fonte aberto e a possibilidade de incorporar novas extensões, adicionando dessa maneira novas funcionalidades, não foram alheios a essa ascensão fulgurante. Este *browser* disponibiliza algumas extensões *Web*, das quais

¹<http://www.w3.org/XML/Query/test-suite/XQTSReportSimple.html>

se realça o “*Web developer*”, que disponibiliza várias ferramentas para desenvolvimento de aplicações *Web*.

Kate

O *Kate*, acrónimo de *Kde advanced text editor*, é um editor de texto do KDE, ambiente gráfico para sistemas UNIX, que fornece um vasto leque de funcionalidades tais como sintaxe *highlighting* para diversas linguagens de programação e *scripting* e uma *shell* integrada, entre outras.

Coordenadas

Para obter coordenadas de locais para o desenvolvimento e testes, recorreu-se ao site “Google maps latitude, longitude Popup”², que permite obter no mapa, as coordenadas de um ponto seleccionado. Essas coordenadas estão no formato decimal e foi necessário convertê-las para o sistema de graus, minutos e segundos. Para esse efeito foi elaborada uma página Web com Javascript embebido para converter coordenadas entre os diferentes sistemas.

4.3 Desenho do protótipo

Para testar as funções propostas neste trabalho foi desenvolvido uma aplicação *Web* com a organização a representada no diagrama no diagrama de casos de uso (figura 4.1) Neste sistema, os actores são o administrador, o agente turístico e o turista, que podem executar as seguintes operações:

Como já foi referido, os actores do sistema são o administrador, o agente turístico e o turista, que podem executar as seguintes operações:

gestão do sistema – o actor desta operação é o Administrador do sistema.

iniciar a sessão – esta operação é realizada por todos os actores, de forma a ficarem autenticados e puderem utilizar do sistema.

terminar a sessão – operação realizada por todos os actores do sistema.

recuperar *password* – operação pode ser realizada pelo agente turístico ou pelo turista e permite recuperar uma *password* esquecida.

visualizar conta ou perfil – os actores desta operação são o agente turístico e o turista permite visualizar os dados pessoais dos utilizadores.

alterar *password* – os actores são o agente turístico e o turista que são os utilizadores que podem alterar a respectiva *password*.

²<http://www.gorissen.info/Pierre/maps/googleMapLocation.php>

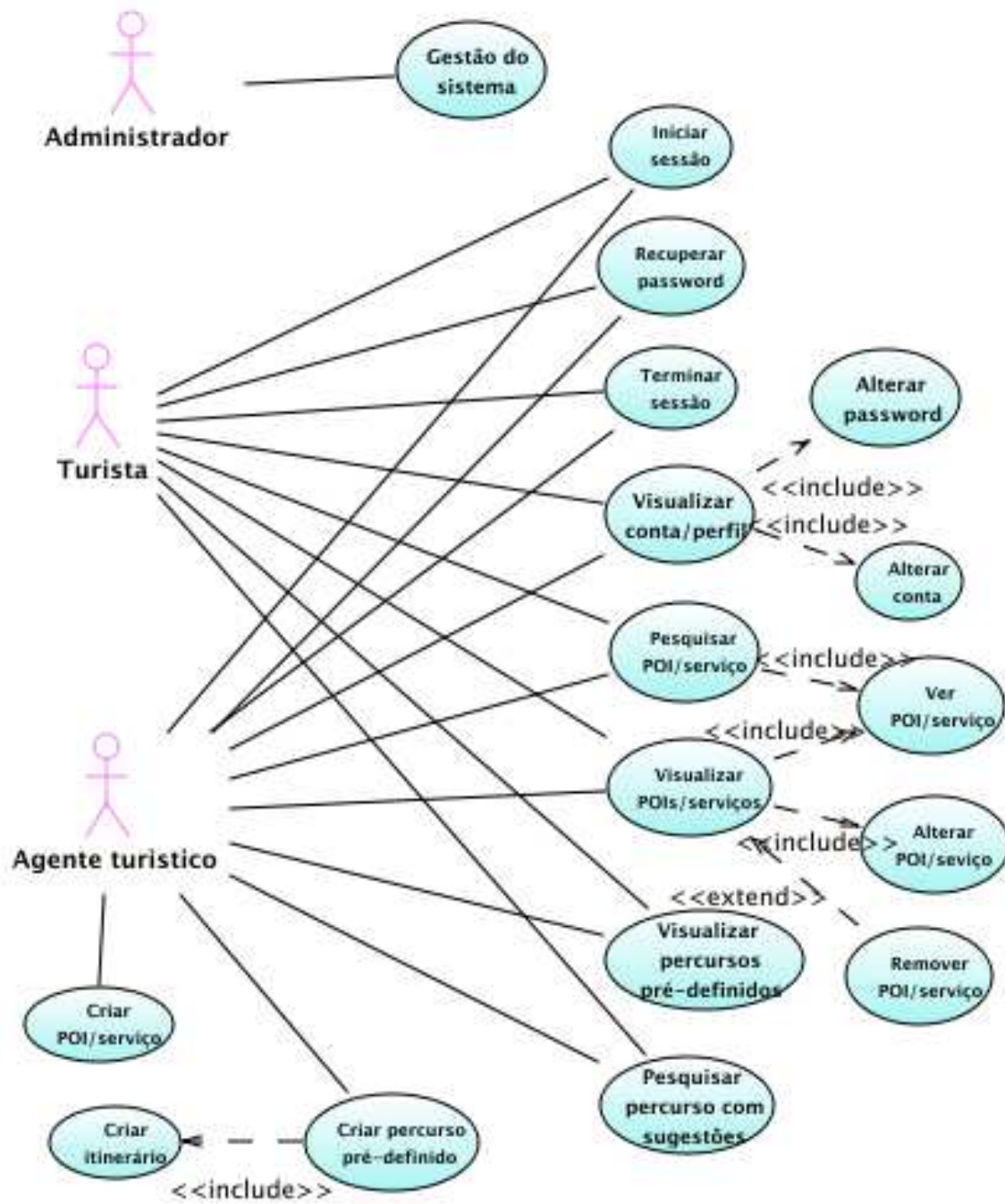


Figura 4.1: Casos de utilização

Alterar conta/perfil – operação que permite actualizar os dados das contas para os agentes e os perfis para os turistas.

pesquisar POI ou serviço – Os actores são os agentes turísticos e os turistas e permite-lhes pesquisar na BD POIs ou serviços segundo critérios geográficos, temporais ou temáticos.

visualizar POIs ou serviços – os actores são os agentes e os turistas e permite-lhes visualizar a informação os POIs ou serviços.

alterar POI ou serviço – o actor desta operação é o agente turístico e permite-lhe manter a informação dos POIs ou serviços actualizada.

visualizar percursos pré-definidos – os actores desta operação são os agentes turísticos e os turistas e permite visualizar os percursos.

pesquisar percurso com sugestão – os actores são o agente turístico e o turista e permite-lhes elaborar uma pesquisa onde são sugeridos até 3 temas de POIs ou serviços que se encontram num percurso.

Criar percurso pré-definido – o actor é o agente turístico e permite-lhe elaborar novos percursos.

criar POI ou serviço – o actor é o agente turístico e permite-lhe adicionar um novo POI ou serviço à BD.

Estas operações estão especificadas de uma forma detalhada no Apêndice A. .

4.4 Implementação do protótipo

Cenário 1

Este cenário exemplifica a criação de um percurso pré-definido, em que o utilizador deve especificar uma sequência ordenada de POIs que constituirão o trajecto a percorrer entre um ponto de partida e um ponto de chegada. O utilizador deve começar por especificar a data, a hora e as coordenadas geográficas do local de partida, bem como o meio de transporte a utilizar (a pé ou de automóvel). De seguida, inicia-se um processo iterativo em que o utilizador deve designar sucessivamente os POIs a visitar e o tempo de duração das visitas. O sistema tem duas funções principais neste processo:

1. listar POIs e serviços organizados por categorias;
2. validar as escolhas do utilizador em função das escolhas geográficas e temporais (por exemplo, calcular o tempo de deslocamento entre locais a visitar).

Para o cálculo das distâncias entre dois pontos geográficos, elaborou-se a função

Figura 4.2: Criar Percurso

```
local:distance($lat1 as xs:double?, $long1 as xs:double?, $lat2 as
xs:double?, $long2 as xs:double?) as xs:double?
```

que recebendo as coordenadas (latitude e longitude) de dois pontos calcula a distância entre eles através da fórmula de *Haversine* [30]. O resultado é devolvido em quilómetros. Tendo o valor da distância calcula-se o tempo necessário para percorrê-la, e isso dependerá da velocidade, ou seja, do meio de transporte que deverá ser utilizado. No desenvolvimento do protótipo foram considerados dois meios de transporte: “a pé” e de “automóvel”. A velocidade estimada para quem se desloca a pé é de 5 km/hora e de automóvel de 50 km/hora. Desta forma, o tempo de viagem é dado pela função

```
local:timeSpent($distance as xs:double?, $transporte as xs:string) as xs:double?
```

Essa função recebe a distância a percorrer e o meio de transporte, e devolve um valor real em horas que é necessário transformar para o formato de tempo que se utiliza convencionalmente. A função utilizada para essa conversão é *doubToDur(\$d as xs:double?)* discutida no capítulo anterior. Ao valor devolvido por essa função, é somado a hora de partida, ou no caso de já não ser a primeira iteração, é somada a hora de chegada mais o tempo de visita.

Para validar temporalmente a data e hora, a data é transformada num dia da semana através da função *dayOfDate*. Em seguida a função *dayOpen* permite determinar se o POI está em horário de funcionamento. É então necessário verificar se é possível realizar a visita no tempo indicado para isso soma-se o tempo de visita à hora de chegada ao POI e obtém-se a hora de partida para o próximo POI. Depois de criado o percurso é possível guardá-lo na BD. Estas operações são executadas tantas vezes quantas as iterações do percurso.

Data	dia	Nome	Tipo	Distância	Horário
2008-10-11	sabado	br	Úteis	0.274	00:00-23:59
2008-10-11	sabado	det	Úteis	0.221	09:00-13:00
2008-10-11	sabado	biblioteca	Úteis	0.064	09:00-13:00
2008-10-12	domingo	br	Úteis	0.274	00:00-23:59
2008-10-18	sabado	br	Úteis	0.274	00:00-23:59
2008-10-18	sabado	det	Úteis	0.221	09:00-13:00
2008-10-18	sabado	biblioteca	Úteis	0.064	09:00-13:00
2008-10-19	domingo	br	Úteis	0.274	00:00-23:59

Figura 4.3: Listagem de serviços disponíveis num intervalo de tempo

Cenário 2

Neste caso, o utilizador deve introduzir as coordenadas e o horário de partida e de chegada do percurso. Deve indicar até três categorias de POIs para o percurso assim como o meio de transporte. A aplicação calcula a distância entre esses dois pontos, e o tempo de percurso. Depois para cada POI irá calcular as coordenadas do ponto da recta mais próximo ao POI. Com essas coordenadas, é calculada a distância POI ao trajecto e o tempo necessário para percorrer esse trajecto. É criada uma lista com todos os POIs que são temporalmente viáveis. O utilizador pode então seleccionar os POIs que considera mais interessantes para incluir no percurso. Para calcular as distâncias dos POIs ao segmento de recta formada pelos pontos coordenados recorreu-se às funções

```
local:u($x1 as xs:double, $y1 as xs:double, $x2 as xs:double, $y2 as xs:double,
    $x3 as xs:double, $y3 as xs:double) as xs:double
```

Esta função calcula o valor de u da fórmula 4.1 explicada à frente. Este valor tem de ser calculado para cada POI. Depois para obter as coordenadas do ponto da recta é chamada a função

```
local:coordinate($x1 as xs:double, $x2 as xs:double, $u as xs:double) as xs:double
```

com as componentes $(x1, x2)$ e $(y1, y2)$ como na formula 4.2. Com essas coordenadas, já é possível obter a distância do POI à recta assim como o tempo necessário a percorrer essa distância. Em seguida são feitas as validações temporais, como no cenário 1. Este processo consegue responder a casos como o 2.(b) e 2.(c).

Desenvolveu-se ainda uma função

local:allOpen(\$cat as xs:string, \$data1 as xs:date, \$data2 as xs:date, \$rest as xs:string,
\$transport as xs:string) as node()*

Esta função permite dada uma localização, categoria, data inicial, data final, restrição temporal e um meio de transporte, obter todos os POIs da categoria \$cat abertos no elemento de tempo $\{[\$data1, \$data2] \cap \$rest\}$. A variável \$rest é a restrição temporal, e pode ser, por exemplo, segunda e quinta. Neste caso, se o meio de transporte for “a pé” serão considerados POIs até distâncias de 1 Km e senão a distância será de 10 Km. O caso 2.(d) fica resolvido com esta função (figura 4.3).

cenário 3

O caso 1.(a) é mais simples e pode ser resolvido recorrendo a instruções XPath:

```
doc("serviços.xml")/SERVICE[name="Gruta"and category="Restaurants"]/latitude.  
doc("serviços.xml")/SERVICE[name="Gruta"and category="Restaurants"]/longitude.
```

O caso 1.(b) mereceu mais esforço, e primeiro calcula-se para cada restaurante a distância ao ponto de referência assim como a validação temporal. É criado um ficheiro auxiliar onde é guardado por ordem crescente de distâncias a identificação dos POIs ou serviços, a distância ao ponto de referência e se está aberto. Em seguida consulta-se o ficheiro XML auxiliar, percorrendo-o até encontrar um POI ou serviço aberto. Esse trabalho é executado pela função

local:search() as node()*

que obtém directamente as variáveis do ponto de referência, categoria, tipo, data ou dia da semana, hora. Ou seja, esta função devolve sempre o POI ou serviço aberto mais próximo.

Funções adicionais

- coordenadas – as coordenadas na forma latitude e longitude em graus, minutos e segundos (exemplo latitude: 37°13'15"N longitude: 8°41'33"), foram convertidas para uma representação decimal para facilitar a sua utilização. Para tal foi criada uma função na linguagem XQuery com o protótipo:

local:convCoor(\$i as xs:string?) as xs:double?

Esta função, dadas as coordenadas atrás referidas, como string, transforma-a em um valor decimal.

- Cálculo da distância entre dois pontos coordenados – dado a forma do planeta Terra ser geóide, foi considerada nesta dissertação como sendo um globo. Para esse cálculo recorreu-se a fórmula de Haversine em [30] que dados dois pontos coordenados calcula a distância entre eles recorrendo à formula de *Haversine* [30].

- Dados um segmento de recta definido por dois pontos A e B , e um terceiro ponto P que não lhe pertence, para calcular as coordenadas do ponto C da recta, mais próximos do ponto P , recorre-se à geometria planar (figura 4.4). Seja o segmento de recta \overline{AB} , com $A = (x_1, y_1)$ e $B = (x_2, y_2)$ e o ponto $P = (x_3, y_3)$,

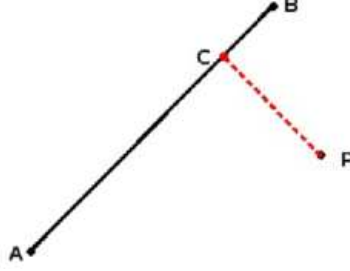


Figura 4.4: Distância de um ponto a uma recta

$$u = \frac{(x_3 - x_1)(x_2 - x_1) + (y_3 - y_1)(y_2 - y_1)}{\|B - A\|} \quad (4.1)$$

e a partir de

$$\begin{cases} x = x_1 + u(x_2 - x_1) \\ y = y_1 + u(y_2 - y_1) \end{cases} \quad (4.2)$$

Obtém-se o ponto $C = (x, y)$ do segmento de recta, mais próximo do ponto P . No caso de $u < 0$, o ponto C é coincidente com A e no caso de $u > 1$ o ponto C será coincidente com B . Conhecendo esse ponto, é fácil determinar a distância desse ponto ao ponto P . Existe, no entanto, uma observação a ter em atenção, como se recorre a geometria planar e visto a terra não ser plana, esta aproximação será válida apenas para distâncias relativamente curtas, na ordem das dezenas de quilómetros.

O Apêndice B apresenta uma breve descrição das funções utilizadas para a implementação deste protótipo.

4.5 Avaliação do sistema

Nas tabelas 4.1 e 4.2 é feita a comparação do protótipo com os guias turísticos apresentados em 2.4. A principal característica que diferencia o sistema proposto neste trabalho em relação aos outros guias, é a possibilidade de criar percursos pré-definidos. Isto permite ao utilizador planear as suas viagens e aferir com antecedência a viabilidade das mesmas.

O CityHelp também suporta o turista na sua fase de planeamento, mas esse apoio limita-se a apresentar caminho entre dois locais, ou seja, funciona como um roteiro e não permite saber se o POI está aberto ou fechado. Em termos de conteúdos, o sistema desenvolvido encontra-se a par com os seus concorrentes apresentando informação sobre os

POIs ou serviços e a possibilidade de visualizar as suas localizações no mapa. Disponibiliza ainda imagens, vídeos e ligações a sítios da Internet dos POIs e onde os utilizadores podem obter mais informação. Em termos de autonomia usufruiu da mesma vantagem do cityHelp e só necessita de se ligar para fazer a actualização da informação ou no caso de querer aceder a uma página Web. Esta característica torna-se desvantagem no sentido de não facultar serviços de pagamentos como é o caso do Cruso. Como o COMPASS dá suporte ao perfil do utilizador no entanto, a sua actualização só é efectuada pelo próprio turista. Em termos de contexto, não chega a ser tão completo como o CRUMPET, não dá suporte ao contexto do dispositivo nem ao contexto da rede.

Guias Turísticos	Origens	Arquitectura	suporte	Objectivo Principal	Aplicação
COMPASS	sistema de recomendação dependente do contexto	interna externa	PDAs	sistema integrado baseado no contexto e sistemas de recomendação em aplicações móveis de turismo	guia turístico baseado no contexto
CRUMPET	serviços móveis personalizados para turismo	interna externa	PDAs	suporte a utilizadores móveis com serviços	guia turístico baseado na localização
Cruso	serviços baseados na localização	interna	Hardware próprio	gerar informação turística baseada no contexto	guia turístico baseado na localização
CityHelp	serviços baseados na localização	interna	PDAs e telemóveis	suporte a visitante em cidades	roteiro turístico personalizado
Sistema proposto	serviços baseados na localização	interna externa	PDAs	suporte a visitantes baseado na localização com suporte temporal	guia turístico baseado na localização e percursos pré-definidos

Tabela 4.1: Comparação geral entre guias turísticos móveis e o protótipo

4.6 Sumário

Uma das principais contribuições deste projecto foi a inclusão do tratamento de informação temporal associada aos POIs e serviços de interesse turístico. Esse tratamento temporal baseou-se principalmente no trabalho de Grandi e teoria de BDT. A teoria de BDT permite a implementação de relações temporais, conhecidas como as relações de Allen (tabela 2.1 ou tabela 2.2) e a representação de instantes, intervalos ou elementos de tempo. Contribuiu ainda na elaboração dos percursos pré-definidos, pois os sistemas SLB normalmente não proporcionam esta funcionalidade. Desta forma, fez-se uma primeira abordagem à construção de percurso turísticos, tendo em consideração os critérios geográficos associados um trajecto e não um ponto fixo, a disponibilidade temporal dos recursos, como por exemplo os horários de funcionamento, e o tempo de deslocamento.

Guia	Obtenção Posição	Ciclo de vida			Tipo de informação	Percurso	Suporte de navegação	Contexto				
		Antes	Durante	Depois				Localização	Tempo	Dispositivo	Rede	Utilizador
COMPASS	gps redes móveis		x		Informação multimédia (texto, imagens) sobre os POIs tais como museus, monumentos, etc. Estes são marcados no mapa como símbolos. Os POIs podem facultar novos serviços		posição do utilizador no mapa	x	x			x
CRUMPET	GPS		x		informação multimédia (texto, imagens e vídeos) sobre POIs tais como restaurantes, hotéis na forma de serviços de recomendações ou dicas pró-activas. POIs da categoria seleccionada são exibidas no mapa como símbolos		o caminho mais curto entre a posição do utilizador e o POI	x		x	x	x
Cruso	GPS e bússola electrónica		x		informação multimédia (texto, imagens e áudio) sobre POIs tais como monumentos, restaurantes. Serviços como bilhetes e pagamentos electrónicos	pré-definido	posição do utilizador	x		x		
cityHelp	GPS ou manual	x	x		informação multimédia (texto e imagens)		caminho mais curto	x				
sistema proposto	GPS ou manual	x	x		informação multimédia (texto e imagens) sobre POIs tais como restaurantes, hotéis, museus, etc... São apresentados no mapa como ícones	percursos pré definidos		x	x			x

Tabela 4.2: Características gerais dos guias turísticos móveis e do protótipo

Capítulo 5

Conclusões

O desenvolvimento tecnológico no domínio das novas tecnologias da informação e comunicação tem aberto com vasto leque de oportunidades para a criação de novos serviços. Nesse contexto, os LBS têm assumido um papel relevante em diversos domínios de aplicação, nomeadamente no Turismo. A integração de LBS nos SI turística está a dar os primeiros passos e certamente, existirão desenvolvimentos significativos nesta área no futuro próximo.

Assim sendo, nesta dissertação foi feito um estudo tendo em vista a implementação de um servidor de informação turística orientado para dispositivos móveis.

O trabalho iniciou-se com uma pesquisa sobre serviços baseados na localização e serviços baseados no contexto e fez-se o levantamento de guias turísticos móveis. Verificou-se que existem diversas abordagens com objectivos diferentes. O armazenamento e a manipulação de informação temporal são lacunas das implementações existentes. Assim, foi efectuado uma análise das bases de dados temporais e, especialmente das bases de dados em XML com suporte temporal, linguagem cada vez mais utilizada em ambientes *Web*.

Fez-se o levantamento dos requisitos funcionais e não funcionais que permitiram elaborar uma estrutura do servidor. Elaborou-se o desenho da arquitectura a implementar e foram escolhidas as ferramentas a utilizar. Foi necessário uma aprendizagem das linguagens envolvidas (HTML, XML, XPath, XQuery e Javascript). Seguiu-se a implementação da aplicação que foi testada tendo em consideração diferentes cenários de utilização.

Estes cenários de utilização mostram que o modelo de dados temporal utilizado é adequado para cobrir os principais requisitos de um SI turística. No futuro, seria interessante incluir no modelo a capacidade de definir períodos de indisponibilidade dos serviços, por exemplo, períodos de encerramento para férias ou por motivo de obras.

Neste trabalho também foi feita uma primeira abordagem ao desenvolvimento de opções específicas para a elaboração de percursos turísticos. Foram apresentados diferentes cenários de utilização da informação disponibilizada pelo sistema e implementadas funções para a concretização desses cenários. No futuro seria interessante a preparação de um con-

junto exaustivo de testes que permitisse avaliar de forma sistemática as soluções propostas, incluindo, por exemplo, testes de funcionalidade e usabilidade.

Desta forma, seria possível realizar uma avaliação mais profunda das soluções propostas, tendo em vista a melhoria das soluções apresentadas.

Apêndice A

Descrição dos Processos

Para a realização do tratamento das operações temporais são necessários os seguintes processos:

addHoursField – Dado uma tabela do horário incrementar dois campos para adicionar um intervalo de horas ao horário.

subHoursfield – Dado uma tabela do horário remove um intervalo de horas representado por dois campos.

addDay – Adiciona uma tabela ao horário que com 2 campos para um intervalo de dias e 2 campos para um intervalo de horas.

subDay – Remove uma tabela do horário que representa um intervalo de dias com os intervalos de horas incluído nessa tabela.

validHourField – Dado uma hora, valida-a quanto ao formato, devolve verdadeiro se for valida e falso em caso contrário.

validTimetable – Dado um horário, verifica a sua validade, devolve verdadeiro em caso afirmativo e falso em caso contrário.

validField – Dado um campo obrigatório, verifica esse campo esta preenchido, se for o caso devolve verdadeiro caso contrário devolve falso.

validCoordinate – Dado uma coordenada verifica se foi introduzida correctamente devolvendo verdadeiro e falso em caso contrário.

bissextile – Dado um ano, verifica se é bissexto, se for devolve verdadeiro senão devolve falso.

No lado da BD serão necessários processos para o tratamento dos dados temporais e suas validações:

dayOfDate – Dado uma data, devolve o dia da semana correspondente, por exemplo, “2008-09-14” obtém-se “domingo”.

containDay – Dados um intervalo de dias da semana verificar se um outro dia está nesse, intervalo, por exemplo, o intervalo de “segunda” à “sexta” verifica se “quarta” está nesse intervalo.

containHour – Dado um intervalo de horas e um instante verifica se esse instante está no intervalo.

doubToDur – Dado um número real este é convertido para uma representação de períodos de tempo.

presentTimeTable – Dado um elemento POI ou serviço apresenta no ecrã o seu horário.

printPOI – Dada a identificação de um POI ou serviço apresenta-o no ecrã.

present – Apresenta todos os POIs e Serviços no ecrã.

CreateElement – Lê os dados do *browser* e cria um novo elemento.

addElement – Dado um elemento POI ou serviço guarda-o na BD.

delElement – Dado o ID de um POI ou serviço, remove-o da BD.

convCoord – Dada uma coordenada na forma graus minutos segundos converte-a para a forma decimal.

doubToCoord – Dada uma coordenada em decimal converte-a para a forma graus minutos segundos.

distance – Dadas as coordenadas de dois pontos calcula a sua distancia em km.

closest – Dados dois pontos coordenados, o meio de transporte e as preferências devolve os POIs ou serviços mais próximos e ao alcance do meio de transporte utilizado no trajecto formado pelas coordenadas.

timeSpent – Dada uma distância, calcula o tempo de percurso mediante o meio de transporte utilizado.

storePerc – dado um percurso armazena-o na BD.

delPerc – dado um percurso apaga-o da BD.

printPerc – apresenta um Percurso no ecrã.

search – Dados as coordenadas de uma posição de referência, o meio de transporte, uma data ou um intervalo de datas e uma restrição de dias da semana, devolve todos os POIs ou serviços que ao alcance do meio de transporte, no intervalo ou dia considerado.

Apêndice B

Casos de Utilização

Este apêndice faz uma apresentação detalhada dos casos de uso introduzidos na secção 4.3

Iniciar sessão: Os actores são o administrador, o agente turístico e o turista e como pré-condição os actores devem estar registados no sistema. A sequência típica de eventos é:

1. O actor insere o seu login e a sua password nos campos adequados no interface.
2. O sistema valida os dados inseridos, apresenta o ecrã de sucesso na autenticação do actor e apresenta o ecrã principal da aplicação onde aparecem as opções disponíveis.

A sequência alternativa é:

2. Os dados inseridos pelo actor são inválidos.
 - (a). O sistema apresenta mensagem de erro e apresenta novamente o interface de inicio de sessão.

Terminar sessão: Os actores são o administrador, agente turístico e turista e como pré-condição os actores devem estar autenticados e ter uma sessão activa. A sequência típica de eventos é:

1. O actor escolhe a opção “terminar sessão” na interface.
2. Sistema pede confirmação.
3. Actor confirma.
4. Sistema termina sessão.

A sequência alternativa é:

3. Cancelar a operação
 - (a). Actor cancela operação.
 - (b). Sistema cancela e termina a operação.

Recuperar password: Os actores são o agente turístico e turista. A sequência típica de eventos é:

1. O actor insere o seu nome de utilizador no campo adequado na interface.
2. O sistema gera uma nova senha e envia-a para o seu email.

Vizualizar conta: Os actores são o agente turístico e o turista. A sequência típica de eventos é:

1. É apresentado ao actor toda a informação sobre a sua conta. Assim como as opções para as operação de alterar dados e alterar password.
2. O actor regressa ao ecrã principal seleccionando a opção “Voltar”.
3. O sistema termina o caso de uso e regressa ao ecrã principal.

As sequências alternativas são:

2. O actor decide actualizar os seus dados.
 - (a1). O actor selecciona “alterar dados”.
 - (a2). O sistema chama o caso de uso “alterar dados”.
2. O actor deseja alterar a password.
 - (b1). O actor selecciona alterar password.
 - (b2). O sistema chama o caso de uso alterar password

Alterar conta: Os actores são o agente turístico e turista e como pré-requisito o actor deve estar a visualizar os dados da conta. A sequência típica de eventos é:

1. O sistema apresenta os dados da conta em forma de formulário e os campos tornam-se editáveis.
2. O actor actualiza os campos que deseja com a excepção do seu nome de utilizador.
3. O utilizador selecciona a opção guardar dados.
4. O sistema valida os campos e guarda a informação na BD.

As sequências alternativas são:

2. O actor cancela a operação.
 - (a1). O actor selecciona a opção cancelar.
 - (a2). O sistema cancela a operação e regressa ao ecrã de visualização de dados.
2. Dados inválidos.
 - (b1). O sistema informa o actor e solicita que introduza novamente os dados.

Alterar Password: Os actores são o administrador, agente turístico e o turista e como pré-condição os actores deve estar a visualizar os dados da sua conta. A sequência típica de eventos é:

1. O actor insere a nova password, a confirmação e a password actual nos devidos campo na interface.
2. O sistema confirma a alteração da password e regressa ao ecrã de visualização de dados.

As sequências alternativas são:

1. Anulação da operação.
 - (a1). O actor decide cancelar a operação através da opção adequada de interface.
 - (a2). O sistema cancela a operação.
2. A Password actual está errada.
 - (a1). A password actual inserida está errada.
 - (a2). O sistema apresenta uma mensagem de erro e solicita nova introdução da senha actual.
2. (b) Nova password não cumpre os critério de segurança.
 - (b1). O actor inseriu uma nova senha que não satisfaz os critérios de segurança necessários.
 - (b2). O sistema indica que a senha inserida não é válida e regressa ao ponto 1.
2. (c) A nova senha não corresponde à confirmação.
 - (c1). A confirmação da nova senha inserida não corresponde à nova senha inserida pelo actor.
 - (c2). O sistema indica a inconsistência entre a nova senha e a sua confirmação e solicita ao actor que as volte a introduzir assim como a password actual.

Criar Poi/serviço: O actor é o agente turístico. A sequencia típica de eventos é:

1. O Actor introduz o nome, a categoria, o tipo, a morada, a coordenadas da localização, a descrição, os detalhes, o contacto, o horário, o endereço Web, uma foto e um vídeo do POI ou serviço.
2. O actor valida o formulário.
3. O sistema verifica a validade da informação e atribui-lhe um identificador e guarda-o na BD.

As sequências alternativas são:

1. (a). O actor cancela a operação.
 - (a1). O sistema termina a operação e regressa ao ecrã principal.
- (b) Introdução de informação multimédia.
 - (b1). O actor selecciona “adicionar foto” ou “adicionar vídeo”.
 - (b2). O sistema pede o caminho da nova foto ou video.
 - (b3). O Actor introduz o caminho.
 - (b4). O sistema adiciona a foto ou vídeo à BD e actualiza formulário.
 - (b4). i) O ficheiro não cumpre as normas do sistema(tamanho demasiado grande ou formato desconhecido).
 - (b4). ii) Sistema informa utilizador que a operação foi realizada com sucesso e regressa ao ponto (b2).
- (c) Introdução de informação temática adicional que depende da categoria ou tipo.
 - (c1). O actor preenche os campos que possam aparecer relativamente a categoria ou tipo de POI ou serviço. Continua no ponto 1.
- (d) Dados inválidos ou em falta.
 - (d1). O sistema informa o actor que existe um ou mais campos em falta ou preenchidos incorrectamente e solicita o actor o corrija.
 - (d2). Regressa ao ponto 1.

Remover POI ou serviço: O actor é o agente turístico. Como pré-requisito, o actor deve estar a visualizar os POIs ou Serviços. A sequência típica de eventos é:

1. O actor selecciona “apagar POI ou serviço”.
2. O sistema solicita o identificador do POI ou serviço.
3. O actor introduz o identificador e valida.
4. O sistema remove o registo da BD e informa o actor do sucesso da operação.

A sequência alternativa é:

3. O actor cancela a operação.
 - (a1). O sistema cancela a operação e regressa ao ecrã principal.

Visualizar POIs e Serviços: O actor é o agente turístico e como pré-requisito deverá existir pelo menos um POI ou serviço. A sequência típica de eventos é:

1. O sistema apresenta a lista de todos os POIs e eventos da BD. A informação apresentada é o identificador, nome, categoria, tipo e zona para cada POI ou serviço.
2. O actor pode alterar a ordenação por diferentes critérios(categoria, tipo, zona).
3. O sistema reordena a lista consoante o critério escolhido pelo actor.

A sequência alternativa:

2. Ver POI ou serviço.
 - (a1). O actor selecciona um POI ou serviço. e escolhe a opção “ver”.
 - (a2). O sistema avança para “Ver POI ou Serviço”.

Ver POI ou serviço: Os actores são o agente turístico e o turista como pré-requisito o actor deve visualizar todos os POIs ou serviços. A sequência típica é:

1. O sistema apresenta toda a informação relativa ao POI ou serviço.
2. O actor visualiza a informação e selecciona voltar.
3. O sistema regressa ao caso de uso “visualizar POIs ou serviço”.

A sequência alternativa é:

2. Alterar POI ou Serviço.
 - (a1). O actor (no caso, o agente de turismo) selecciona alterar.
 - (a2). Sistema chama caso de uso “alterar POI ou serviço”.

Pesquisar POI ou serviço: Os actores são o agente turístico e o turista. A sequência típica de eventos é:

1. O Actor introduz a localização de referência, data e horas de partida, o meio de transporte, a categoria ou tipo do POI ou serviço que procura.
2. Actor valida a pesquisa.
3. O sistema apresenta os registos com localização no raio de acção do actor, válidos temporalmente. A informação apresentada é a localização, o horário, os contactos, assim como a distância a percorrer e uma estimativa do tempo necessário ao trajecto.

As sequências alternativas são:

1. 2. O actor cancela a operação.
 - (a1). O sistema termina a operação e regressa ao ecrã principal.
1. Introdução de dados actuais.
 - (b1). O actor selecciona a opção “Utilizar dados actuais”.
 - (b2). O sistema actualiza automaticamente os campos da data e das horas com as actuais.
1. Pesquisa não temporal.
 - (c1). O actor selecciona opção “todos” relativa aos POIs ou serviços.
 - (c2). O actor valida a pesquisa.
 - (c3). O sistema apresenta o POI ou serviço mais próximo, independentemente do seu estado de funcionamento (aberto ou fechado).

1. Introdução de um período de tempo.
 - (d1). O actor introduz um período de tempo com data inicial e data final.
 - (d2). O sistema apresenta os POIs ou serviços existentes próximos da posição de referência que satisfaçam as condicionantes temporais.
1. Introdução de elementos de tempo.
 - (e1). O actor introduz um intervalo de tempo assim como os dias da semana dentro desse intervalo em que a pesquisa deve ser realizada.
 - (e2). O sistema apresenta os POIs ou serviços no raio de acção do actor considerados apenas no elemento de tempo introduzido.
3. Os dados iniciais inválidos.
 - (a1). O sistema informa o actor e solicita que introduza novamente os dados.

Criar Percurso pré-definido: O actor é o agente turístico e como pré-requisito o actor deve seleccionar a opção “Criar Percurso pré-definido”. A sequência típica de eventos é:

1. O sistema apresenta um formulário para a inserção de um novo Percurso.
2. Actor cria um novo itinerário.
3. O sistema adiciona a informação ao formulário do novo itinerário (meio de transporte, POIs, duração e mapa).
4. O actor atribui um nome, classifica-o, descreve-o e avalia os custos.
5. O actor insere imagem e um vídeo.
6. O actor selecciona “Guardar Percurso”.
7. O sistema guarda o percurso.

As sequências alternativas são:

2. 4. 5. 6. O actor cancela a operação.
 - (a1). O sistema termina a operação.
3. Dados inválidos.
 - (a1). O sistema informa o actor e solicita que introduza novamente os dados.
4. Inserção de uma nova imagem.
 - (b1). O actor adiciona nova imagem.
 - (b2). O sistema junta a nova imagem.
1. Inserção de vídeo.
 - (c1). O actor adiciona novo vídeo.
 - (c2). O sistema junta o novo vídeo.

Criar itinerário: Os actores são o agente turístico e turista e como pré-requisito o actor deve seleccionar a opção “Criar itinerário”. A sequência típica de eventos é:

1. O sistema apresenta um formulário para a inserção de um novo itinerário.
2. Actor introduz dados iniciais como o local de partida, o meio de transporte a utilizar, a data, hora de partida e valida.
3. O sistema apresenta os dados iniciais.
4. O actor selecciona o POI a introduzir e o tempo de visita e valida.
5. O sistema apresenta numa tabela o POI, a sua localização, hora de chegada, a nova hora de partida, a distância a percorrer da localização anterior e a validade temporal da visita a esse POI.
6. O actor verifica a validade e selecciona “Guardar itinerário”.
7. O sistema guarda o itinerário.

As sequências alternativas são:

2. 4. 6. O actor cancela a operação
(a1). O sistema termina a operação.
3. Dados inválidos.
(a1). O sistema informa o actor e solicita que introduza novamente os dados.
5. O POI introduzido não existe.
(a1). O POI introduzido não existe na BD.
(a2). O sistema informa o actor e volta ao ponto 4.
6. O actor verifica que o POI não é válido temporalmente.
(b1). O actor apaga a última iteração ao seleccionar “apagar último”.
(b2). O sistema remove última iteração e regressa ao ponto 5.
6. O actor quer adicionar um novo POI ao itinerário.
(c1). Regressa ao ponto 4.

Visualizar percursos pré-definidos: Os actores são o agente turístico e o turista e como pré-requisito deve existir pelo menos um percurso. A sequência típica de eventos é:

1. O sistema apresenta uma lista com todos os percursos pré-definidos existentes (o sistema apresenta o nome, o tema e o meio de transporte).
2. Actor pode seleccionar um percurso e ver os detalhes.

A sequência alternativa é:

1. Não existe nenhum percurso.
(a1). O sistema indica que não existe nenhum percurso e regressa ao ecrã principal.

Pesquisar percursos com sugestões: Os actores são o agente turístico e o turista. A sequência típica de eventos é:

1. O Actor introduz a localização de partida e de chegada, data e horas de partida, um meio de transporte, assim como até 3 categorias dos POIs ou serviços que deseja ver aparecer.
2. Actor valida.
3. O sistema apresenta em forma de lista um percurso com todas todos os POIs ou serviços das categorias escolhidas pelo actor. No percurso só aparecem as opções temporalmente válidas, assim como as distâncias a percorrer e os pontos mais próximos do trajecto principal.

As sequências alternativas são:

1. 2. O actor cancela a operação.
 - (a1). O sistema termina o caso de uso e regressa ao ecrã principal.
1. Introdução de dados actuais.
 - (b1). O actor selecciona a opção “Utilizar dados actuais”.
 - (b2). O sistema actualiza automaticamente os campos da data e das horas com as actuais.
2. Os dados iniciais inválidos.
 - (a1). O sistema informa o actor e solicita que introduza novamente os dados.

Bibliografia

- [1] Direcção Geral do Turismo. O turismo em 2006 - Portugal continental e regiões autónomas, consultado em Maio 2008. <<http://www.dgturismo.pt/Português /conhecimento/estudoseestatisticas/Pages/Turismoem.aspx>>
- [2] World Travel & Tourism Council. Tourism Satellite Accounting, the 2008 travel & tourism economic research - Portugal, consultado em Junho 2008, <http://www.wttc.org/eng/Tourism_Research/Tourism_Satellite_Accounting/TSA_Country_Reports/Portugal>
- [3] Eva Milheiro. Quiosques de informação turística como veículos de disseminação da informação, *Investigação em turismo*, 2001, pp. 331–348
- [4] I. Kingsley and D. Fesenmaier. Travel Information Kiosks: An Emerging Communications Channel for the Tourism Industry. *Journal of Travel & Tourism Marketing*, Vol. 4, N° 1, 1995.
- [5] United Nation and World Tourism Organization. Recommendation on Tourism Statistics, *Statistical Papers, Series M*, N° 83, 1994,
- [6] Sílvia Fernandes. Potencial dos sistemas e tecnologias de informação para o negócio turístico, *Revista dos algarves*, nº2, 1997
- [7] M. Goodchild. Integrating GIS and Environmentat Modeling at Global Scales. In *proc GIS/LIS'91*, vol. 1 , pages 117-127, 2001
- [8] Stefan Steiniger, Moritz Neun and Alistair Edwardes. *Introductory lecture on Location Based Services*, 2006
- [9] F. Espinoza, et Al. GeoNotes: Social and Navigational Aspects of Location-Based Information Systems. In: Abowd, Brumitt and Shafer, ed. *Ubicomp 2001: Ubiquitous Computing.*, International Conference, September 30 – October 2, Atlanta, Georgia. Berlin: Springer, pp. 2–17.
- [10] A. J. Brimicombe. GIS - Where are the frontiers now? In: *Proceedings GIS 2002.*, Bahrain, pp. 33–45, 2002

- [11] Open Geospatial Consortium. "Glossary of terms", 24 May 2007. consultado em Março 2008, <http://www.opengeospatial.org/ogc/glossary/1>
- [12] Spatial Media. Glossary, LBSzone.com, consultado em Março 2008 [online] <http://www.lbszone.com/component/option,com_glossary/Itemid,27/>
- [13] K. Varrassi, et Al. Developing GIS-Supported Location-Based Services. In: Proc. of WGIS'2001 – First International Workshop on Web Geographical Information Systems. Kyoto, Japan. pp. 423–432, 2001
- [14] X. Lopez. Location-Based Services. In: Karimi, H. A., Hammad, A., ed. Telegeoinformatics. CRC Press, pp. 171–188, 2004
- [15] Narushige Shioda, et Al. The impact and penetration of location-based services, UCL Eprints, pp. 1–16, 2002
- [16] K. Truong, G. Abowd and J. Brotherton. Who, What, When, Where, How: Design Issues of Capture & Access Application. Proceedings of the International Conference "Ubiquitous Computing" (UbiComp 2001), Berlin, pp. 209–224, 2001
- [17] Carlo Zaniolo, et Al. Advanced Database Systems, Morgan Kaufmann Publishers Inc, USA, 1997
- [18] Andreas Steiner. "A Generalisation Approach to Temporal Data Models and their Implementations", PhD Thesis Departement Informatik, ETH Zurich, November 1997
- [19] James F. Allen, Maintaining knowledge about temporal intervals. Communications of the ACM, vol. 26(11), 1983, pp. 832–843
- [20] P. Tarasewich. Designing Mobile Commerce Applications. Communications of the ACM, Vol. 46, No.12, December 2003
- [21] T. Reichenbacher. Mobile Cartography - Adaptive Visualisation of Geographic Information on Mobile Devices. Verlag Dr. Hut, München, 2004, ISBN 3-89963-048-3
- [22] A. Dey, G. D. Abowd. Towards a Better Understanding of Context and Context Awareness. Technical Report, GIT-GVU-99-22, Georgia Institute of Technology, 1999
- [23] World Wide Web Consortium. Extensible Markup Language (XML), <http://www.w3.org/XML/>, [online] Junho 2008
- [24] Fabio Grandi and Federica Mandreoli. The valid Web: an XML/XSL Infrastructure for Temporal Management of web Documents, pp.1–10, 2000
- [25] Fusheng Wang. XML-Based Support for Database Histories and Document Versions, University of California, Los Angeles, 2004

- [26] s. a. FunctX XQuery Functions, consultado em Março 2008, <http://www.xquery functions.com/xq>, Setembro 2008
- [27] Christoph Grün. Mobile Tourist Guides, Evaluation of the State of the Art and Development of a Light-weight Framework for Location-based Services, Johannes Kepler, Universität Linz, August 2005
- [28] European Media Laboratory. “CRUMPET”, European Media Laboratory, consultado em Março 2008, <http://www.eml-development.de/english/research/crumpet/index.php> [2008]
- [29] Agent for Tourism Mobility, eml Press Release, January [2003]
- [30] R. W. Sinnott, “Virtues of the Haversine”, vol. 68 pg [158], decembre 1984,
- [31] eXist-db, <<http://exist-db.org/>>, [online] Julho 2008